Membership Inference Attacks on Sequence Models

Lorenzo Rossi*, Michael Aerni†, Jie Zhang†, Florian Tramèr†
*CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
†ETH Zurich, Zurich, Switzerland
lorenzo.rossi@cispa.de,
{michael.aerni,jie.zhang,florian.tramer}@inf.ethz.ch

***Abstract*—Sequence models, such as Large Language Models (LLMs) and autoregressive image generators, have a tendency to memorize and inadvertently leak sensitive information. While this tendency has critical legal implications, existing tools are insufficient to audit the resulting risks. We hypothesize that those tools' shortcomings are due to mismatched assumptions. Thus, we argue that effectively measuring privacy leakage in sequence models requires leveraging the correlations inherent in sequential generation. To illustrate this, we adapt a state-of-the-art membership inference attack to explicitly model within-sequence correlations, thereby demonstrating how a strong existing attack can be naturally extended to suit the structure of sequence models. Through a case study, we show that our adaptations consistently improve the effectiveness of memorization audits without introducing additional computational costs. Our work hence serves as an important stepping stone toward reliable memorization audits for large sequence models.**

1. Introduction

Large sequence models, such as LLMs, are trained on a vast portion of the Internet. Since this data contains sensitive or protected information [1, 2], understanding and mitigating memorization of sequence models is an important task. Membership inference attacks (MIAs [3]) are a key tool for this task, which aims to classify whether a given sample was part of the training data. However, current MIAs for large sequence models cannot reliably prove the presence or absence of memorization [4].

In contrast, typical discriminative settings (e.g., image classification) exhibit highly successful membership inference attacks such as the Likelihood Ratio Attack (LiRA) [5]. Conceptually, LiRA first trains shadow models [3] that mimic a victim model and then uses a sample's cross-entropy loss on the victim and shadow models as the input to a statistical hypothesis test.[1] Hence, one might expect the same methodology to yield a strong baseline for large sequence models. However, a naive application can fail to uncover significant memorization, as seen in Figure 1 ("Naive Approach").

1. In practice, the loss is rescaled to better match statistical assumptions.

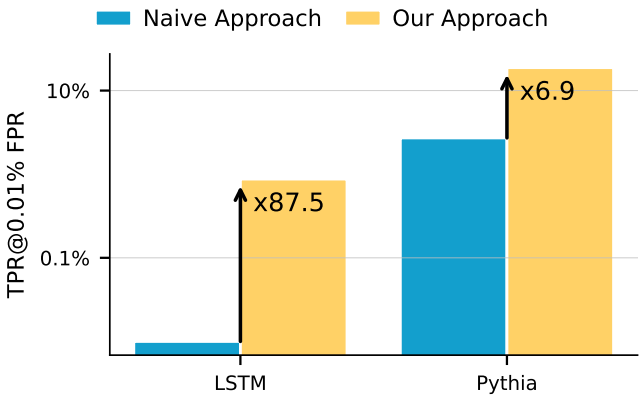


Figure 1: **Strong membership inference for sequence models must consider correlations between sequence elements.** We apply LiRA, a state-of-the-art MIA, to measure memorization in language modeling. First, we use LiRA as-is, where membership guesses only depend on a sample's loss ("Naive Approach"). This approach uncovers only a small amount of memorization, as indicated by TPR (True Positive Rate) values close to the FPR (False Positive Rate) of 0.01%. However, if we adapt LiRA to explicitly consider correlations between sequential predictions ("Our Approach"), we can uncover significantly more memorization (up to 87.5x more). See Appendix A for details.

We hypothesize that the implicit assumptions of that approach are partially invalid for sequence models. In particular, standard MIAs assume that a sample's loss is sufficient to determine membership. However, in the context of sequence modeling, this treats all sequence elements as independent—an assumption that is clearly wrong for domains such as language. We hence argue that strong MIAs for sequence models crucially need to exploit the correlation between sequence elements.

In this paper, we explore adaptations to LiRA that account for within-sequence correlations. By explicitly estimating the covariance between sequence elements in a sample-efficient way, our adaptations uncover significantly more memorization in Figure 1 than the naive approach—with only few shadow models and trivial overhead.

We then study our adaptations on different types of se-

quence models, where we show that those improvements are consistent: accounting for within-sequence correlations can significantly help an attack to uncover memorization, and never truly hurts. Additionally, we compare different types of covariance estimators, and find that a strong inductive bias can sometimes reduce the required number of shadow models by an order of magnitude. Therefore, while we do not propose a new standalone attack, our discoveries serve as an important stepping stone toward truly stronger MIAs for large sequence models.

2. Preliminaries and Related Work

We first describe memorization and membership in general, and then introduce autoregressive models as one specific instance of sequence models.

2.1. Memorization and Privacy Auditing

The extent of memorization in sequence models remains an active area of debate. Although some studies suggest that large-scale models exhibit limited memorization and are unlikely to expose sensitive information under normal conditions [6, 7, 8, 9], other works provide strong evidence that these models can memorize and leak substantial portions of their training data [10, 11, 12, 13, 14, 15, 16, 17]. The discrepancy between these findings is probably due to differences in evaluation methodologies, dataset characteristics, and the selection of test samples (some types of data are more prone to memorization [10, 18]).

Beyond empirical evidence, theoretical studies indicate that memorization is not simply an unintended byproduct of large-scale training, but it can be even necessary to achieve a low generalization error [19, 20]. This raises fundamental questions about the trade-off between model utility and privacy risks. Understanding the mechanisms of memorization, identifying vulnerable data instances, and designing robust privacy auditing techniques are essential to mitigating the risks associated with sequence models.

2.2. Membership Inference Attacks

The goal of MIAs is to determine whether a specific sample was part of the training data [3]. This is typically framed as a standard security game [5, 21, 22], where the objective is to make a binary prediction—indicating whether the sample was included in the training dataset or not. Rather than making strict binary predictions, MIAs often rely on a membership inference score $\mathcal{A}(f, x)$, where f, x are respectively the model and sample audited. A higher scores reflect greater confidence that a sample belongs to the training dataset. A threshold can always be applied to convert these scores back into binary decisions. Using a soft prediction approach enables the evaluation of performance across all possible false positive rates. This, in turn, allows for a comprehensive analysis of the full ROC curve.

When MI is employed as a privacy auditing tool, as is frequently done in computer security [5, 18, 23, 24], these attacks are usually assessed using worst-case metrics such as the attack's true positive rate (TPR) at low false positive rates (FPR). Consequently, if a membership inference attack can reliably and consistently compromise the privacy of even a small subset of vulnerable users in a sensitive dataset, the training algorithm can be considered to leak private information (see [5, 18] for a more detailed discussion).

Following the standard membership inference security game is often impractical, as it would require training too many models to get reasonable estimates for the TPR @ low FPR metric. For this reason, it is common to sample a random subset of the canaries and add them to the training data at each run [5, 18, 25].

Different membership inference attacks were developed specifically for LLMs [26, 27, 28]. Some of the existing black-box attacks exploit the sequential structure of language and construct a statistic by aggregating the per-token scores in a specific way. For instance, [26] showed that only considering the $K\%$ of tokens with the smallest likelihood improves the performance of the attack.

Many works use the same global threshold across all the samples. However, different studies have shown that calibrating the threshold for each sample greatly improves membership inference attacks, as some samples are harder to learn [5, 29, 30]. This is particularly problematic because most of the uncalibrated membership inference attacks are strong *non-membership inference attacks*; they cannot reliably distinguish *easy-to-predict non-members* and *hard-to-predict members* [5, 30]. Several studies address this issue by calibrating predictions [5, 31, 32] using numerous shadow models. While employing a large number of shadow models improves MIA performance, it also comes with a major drawback—significantly increasing computational costs. In the following, we focus on the Likelihood Ratio Attack as a state-of-the-art instance of such attacks.

The Likelihood Ratio Attack (LiRA). [5] developed a method to calibrate the scores using shadow models. The main idea is to train many shadow models with and without the target sample x. Then, compute a score S for each run, and estimate the parameters of a Gaussian distribution for the IN case, where the target sample x was part of the training data, and the OUT case, where it was not part of the training data. Finally, compute the likelihood ratio

$$\mathcal{A}(f, x) := \frac{\mathcal{N}(S(f; x) \mid \mu_{x,\text{in}}, \sigma^2_{x,\text{in}})}{\mathcal{N}(S(f; x) \mid \mu_{x,\text{out}}, \sigma^2_{x,\text{out}})}\,.$$

They also noticed that using scores of multiple augmented versions of the same sample improves the performance. To account for multiple scores, they model each IN and OUT score as an independent Gaussian distribution.

The Neyman-Pearson lemma [33] ensures that the likelihood ratio test is the most powerful test for a given model. In LiRA, this model assumes a Gaussian loss distribution, making the test uniformly most powerful under that assumption. If the true loss distribution deviates from Gaussianity, the resulting likelihood ratio test may be suboptimal in practice, limiting the expressiveness and effectiveness of the attack.

2.3. Autoregressive Sequence Models

Autoregressive models are a common way to solve a wide range of problems across various sequential domains, including text generation [34], image generation [35, 36], time series forecasting [37], and protein structure prediction [38]. The underlying assumption of autoregressive models is that each sequence element only depends on its preceding elements, formalized as follows:

$$x_{t+1} \sim p(x_{t+1} \mid x_1, ..., x_t), \quad t \in \{1, 2, \dots\},$$

where the conditional probability distribution $p(x_{t+1} \mid x_1, ..., x_t)$ models the likelihood of the next element given the past observations. The likelihood of a given sequence is hence the product of individual elements' likelihoods.

Several architectures have been developed for different types of sequential data:

Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM). Recurrent neural networks (RNNs), particularly Long Short-Term Memory (LSTM) networks [39], have historically been used due to their ability to capture long-range dependencies.

Transformer-based architectures and LLMs. Transformer architectures [40] revolutionized autoregressive modeling, particularly in text generation. Unlike RNNs and LSTMs, transformers rely entirely on self-attention mechanisms, allowing them to capture long-range dependencies more effectively. LLMs, such as Pythia [41], build on this foundation, scaling transformers to billions of parameters to achieve human-like text generation capabilities.

Autoregressive Image Generation. Autoregressive models have been extended to image generation by sequentially modeling the dependencies between pixels. For example, PixelCNN++ [36] employs a convolutional architecture that efficiently captures these pixel dependencies while preserving the autoregressive property.

2.4. Covariance Estimation

LiRA models a sample's loss using Gaussian distributions. For autoregressive models, a sample's loss is the sum of per-element losses. An alternative is hence to model the losses of individual sequence elements as a multivariate Gaussian, which requires estimating a covariance.

The simplest approach to estimating the covariance matrix is through the maximum likelihood estimator (MLE):

$$\Sigma_{\text{MLE}} = \frac{1}{N} \sum_{i=1}^{N} (X_i - \bar{X})(X_i - \bar{X})^T,$$

where the $\{X_i\}_{i=1}^N$ are the N observed vectors, and $\bar{X} = \frac{1}{N} \sum_{i=1}^{N} X_i$. In LiRA, N represents the number of shadow models used, and X_i represents the vector of (log) losses of the i-th shadow model. Although the MLE is unbiased, it has notable shortcomings. In particular, it often fails to provide reliable estimates of the eigenvalues of the covariance matrix, and in scenarios where the number of samples is smaller than the number of dimensions, the resulting matrix is non-invertible. This poses significant challenges, especially in high-dimensional applications where an accurate MLE requires many samples.

One straightforward solution is to assume a diagonal covariance structure, which effectively treats different features as uncorrelated. A more advanced type of structure is given by shrinkage-based covariance estimators. These methods compute a convex combination of the MLE-based covariance matrix and a diagonal matrix with controlled variance. The key idea is to balance the variance of the sample covariance with the bias introduced by the target matrix, thereby improving the overall estimation quality.

A particularly effective shrinkage estimator is the Oracle Approximating Shrinkage Estimator (OAS) [42]. OAS improves the estimation of covariance matrices by adaptively selecting the shrinkage intensity based on the characteristics of the sample data. Rather than using a fixed shrinkage parameter, OAS computes the optimal intensity that minimizes the expected error (typically measured by the mean squared error) between the true covariance matrix and the shrinkage estimator. In practice, the OAS estimator constructs the covariance estimate as follows:

$$\Sigma_{\text{OAS}} = (1 - \alpha)\Sigma_{\text{MLE}} + \alpha F, \tag{1}$$

where Σ_{MLE} is the sample covariance matrix estimated via MLE, F is $\frac{\text{tr}(S)}{p} I$, p is the dimensionality of the data, I is the identity matrix, and α is the shrinkage intensity determined from the data.

3. Membership Inference on Sequence Models

In the following, we focus on language models for simplicity and hence use the terms token and sequence elements interchangeably. However, all points apply for more general sequence models, such as autoregressive image generators (see Appendix C for an example). Moreover, we use LiRA as a baseline MIA, but our observations also apply to other shadow model-based attacks such as [32, 43].

3.1. Mismatched Assumptions in Naive LiRA

We hypothesize that the standard LiRA attack has implicit assumptions that can be invalid for sequence models. Concretely, those assumptions are

1) **Independence**: The membership signal is sufficiently captured by a sample's loss. Multiple statistics for a single sample are independent.
2) **Heteroscedasticity**: the variance of a sample's loss (or a derived statistic) depends on whether the sample is in the training data or not.

In the following, we discuss how those assumptions might be invalid for sequence models and propose adaptations.

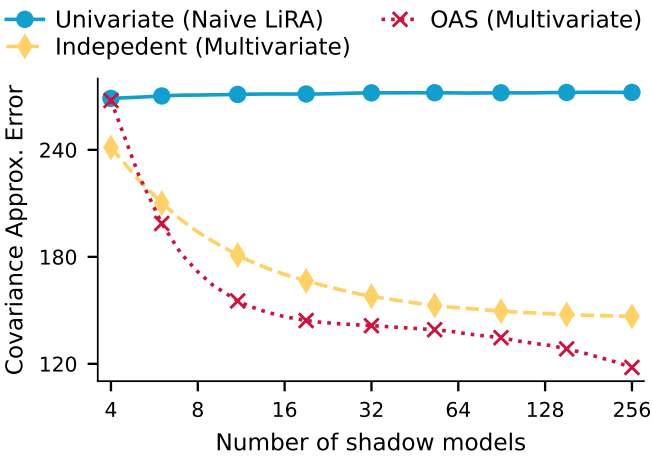


Figure 2: **Assuming independence between per-token losses can yield a large approximation error.** We use the full covariance MLE (with 484 shadow models) as the gold standard, and measure the distance in Frobenius norm ("Covariance Approximation Error") to other models' estimates. As the number of shadow models increases, we analyze this error using 1,000 average-case canaries from the IN case of Pythia 1b with a sequence length of 128. See Figure 6a in Appendix B for the corresponding figure for the OUT case, and for more details, see Appendix A.

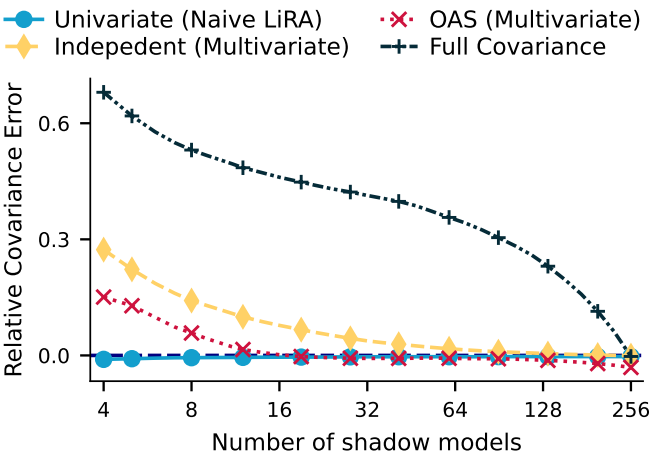


Figure 3: **Assuming shared covariances (homoscedascity) is empirically beneficial.** The Relative Covariance Error is measured as the relative difference between the Covariance Approximation Error of the Class-Wise and Shared covariance estimates. As the number of shadow models increases, we analyze this error using 1,000 average-case canaries from the IN case of Pythia 1b with a sequence length of 128. See Figure 6b in Appendix B for the corresponding figure for the OUT case, and for the experimental details, see Appendix A.

Independence. For traditional classification models, such as image classifiers, a sample's cross-entropy loss depends on the entire sample. In contrast, autoregressive models typically calculate a loss per token, such that a sample's loss is the sum (or mean) of per-token losses. Hence, a naive application of LiRA—which only uses the loss of the entire sequence—corresponds to modeling the token losses as independent Gaussians.

However, the per-token losses of autoregressive models are highly dependent by definition! We hence model the entire per-token loss vector as a multivariate Gaussian, thereby explicitly accounting for inter-token correlations. This results in the following modified hypothesis test, where the IN and OUT distributions are now multivariate Gaussians:

$$\mathcal{A}(f,x) := \frac{\mathcal{N}(\mathbf{S}(f;x) \mid \mu_{x,\text{in}}, \mathbf{\Sigma}_{x,\text{in}})}{\mathcal{N}(\mathbf{S}(f;x) \mid \mu_{x,\text{out}}, \mathbf{\Sigma}_{x,\text{out}})},$$

where the score function $\mathbf{S}(\cdot,\cdot)$ applies element-wise for every per-token loss, $\mu_{x,\text{in}}, \mu_{x,\text{out}}$ are mean vectors with dimensionality equal to the sequence length T, and $\mathbf{\Sigma}_{x,\text{in}}, \mathbf{\Sigma}_{x,\text{out}}$ are $T \times T$ covariance matrices.

We illustrate this model mismatch in Figure 2 for sequences of 128 tokens. First, we estimate the full covariance matrix using many shadow models (484) for an LLM (Pythia 1b). We then compute the covariance matrices according to the original LiRA model ($\mathbf{\Sigma} = \sigma^2\mathbf{I}$; "Naive LiRA (Univariate)") and a model that estimates per-token loss variances but no correlations ($\mathbf{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_T^2)$; "Independent (Multivariate)").[2] Finally, we compute the distance of the full covariance matrix to the two approximations in terms of Frobenius norm. Both approximations do not converge to the full covariance, and have a large approximation error—especially for few shadow models.

Unfortunately, the full covariance matrix has a parameter count that is quadratic in the number of tokens. This significantly increases the sample complexity required for accurate covariance estimation. Because the number of samples corresponds to the number of shadow models, the larger sample complexity also increases computational cost.

To obtain the richer model with a manageable computational cost, we explore efficient covariance estimation techniques. The OAS estimator [42] is particularly well-suited for high-dimensional settings with limited samples. As seen in Figure 2 ("OAS (Multivariate)"), OAS approximates the full covariance more accurately with few shadow models, and eventually matches the full covariance.

Heteroscedasticity. Standard LiRA assumes that the covariance of losses differs for in vs. out distributions (heteroscedasticity). However, for sequence models, we find that using the same covariance matrix for both distributions (homoscedascity) is empirically more accurate. This effectively doubles the number of available samples for covariance estimation, as we use the shadow models from both the in and out distributions to estimate the covariance matrix.

Concretely, we use 484 shadow models to obtain an accurate estimate of the full covariance matrices to serve as a gold standard. We then estimate covariances with different structures for a varying number of shadow models, once class-wise (homoscedastic), and once shared between classes (heteroscedastic). We then calculate the approxima-

2. We drop the in/out suffix for brevity.

tion error in terms of Frobenius norm between the estimates and the gold standard, and plot the relative difference (class-wise error minus shared error, normalized by the shared error). A large relative difference indicates that the class-wise covariance estimate has a much larger approximation error than the shared covariances.

Figure 3 illustrates the benefits of using the same in and out distributions. We find that using the same covariance for in and out distributions is typically beneficial, as indicated by the mostly positive relative difference. Those benefits are particularly pronounced when the number of shadow models is small; we conjecture that this is again due to the reduction in sample complexity.

3.2. Attack Variants

Based on the previous observations, we consider the following concrete adaptations for LiRA on sequence models:

- **Univariate:** A baseline approach where per-token scores are averaged into a single scalar, and a univariate Gaussian distribution is estimated for each class (member and nonmember). This corresponds to a naive application of LiRA to sequence models.
- **Independent:** Uses per-token scores directly but retains the independence assumption, estimating only the diagonal tokens of the covariance matrix.
- **OAS:** Drops the independence assumption by estimating the covariance matrix using OAS.

Additionally, based on the Heteroscedasticity assumption, each of these methods can be further divided:

- **Class-Wise:** Estimates separate covariance matrices for members and nonmembers.
- **Shared:** Uses a single covariance matrix for both classes, leveraging all available samples for estimation. In this case, we first center the members and nonmembers using per-class means, but then estimate a shared covariance matrix.

In some cases, tokens in the sequence may be redundant or introduce unnecessary noise. To address this, we explore methods to reduce the sequence length while preserving essential information:

- **Group.** The sequence is divided into fixed-size chunks, and the average score is computed for each chunk. This approach smooths local variations while retaining overall trends.
- **Min.** The token with the smallest score is selected. This follows a similar intuition as Min-K%, emphasizing the most confident (least anomalous) tokens in the sequence.
- **Max.** Analogous to Min, but instead selects the token with the largest score, capturing the most uncertain or anomalous regions in the sequence.

4. Experiments

4.1. Setup

We study attack variants across two language modeling tasks: training an LSTM from scratch and fine-tuning a transformer-based LLM. This comparison provides insights into how different generative models memorize and expose sensitive information under various conditions. Furthermore, in Appendix C, we describe a case study using PixelCNN++ [36], an autoregressive image generator. Our objective is to assess the performance of membership inference attacks (MIA) under different model architectures and training paradigms.

We use two types of canaries: average-case and worst-case. We sample average-case canaries uniformly from the test set. They hence represent naturally occurring sequences within the dataset and help measure memorization under typical conditions. For worst-case canaries, we aim to pick samples that are particularly prone to memorization. Following common wisdom [6, 44], we use random token sequences as worst-case canaries. These synthetic sequences serve as adversarial inputs designed to maximize memorization effects, thereby approximating an upper bound of privacy leakage. In both cases, we consider 10,000 canaries.

We train all the language models on the PersonaChat dataset [45], which consists of conversations of people describing themselves. This dataset mimics a realistic setting where privacy leakage could be a concern. For each architecture and canary type, we train a total of 64 models, ensuring that each canary appears in exactly half of the model's training data. We then adopt a common leave-one-out strategy for evaluation. That is, we treat each of the 64 models as a target and use the remaining 63 models as shadow models.

LSTM-Based Models. We train the LSTM models [39] from scratch using a standard architecture. The architecture consists of an initial embedding layer to encode tokens, followed by two LSTM layers with a hidden dimension of 192 and a fully connected prediction head for token classification. We use the Pythia 1b tokenizer and train the model on the standard next-token prediction task with cross-entropy loss. See Table 1 from Appendix D for the specific choice of hyperparameters.

Transformer-Based Models (Pythia 1b). For both average- and worst-case canaries, we fine-tune the deduplicated Pythia 1b model [41], a 1-billion-parameter transformer-based language model pre-trained on a large corpus of data from the internet. The training follows the standard next-token prediction objective using cross-entropy loss. To further explore the importance of the number of shadow models, we fine-tune a large number (484) of shadow models using average-case canaries. As in the other settings, we run leave-one-out on all the 484 shadow models. See Table 1 from Appendix D for specific hyperparameters.

Evaluation. We focus the evaluation on the true positive rate (TPR) in the low false positive rate (FPR) regime, specifically on TPR@0.01% FPR. This metric identifies

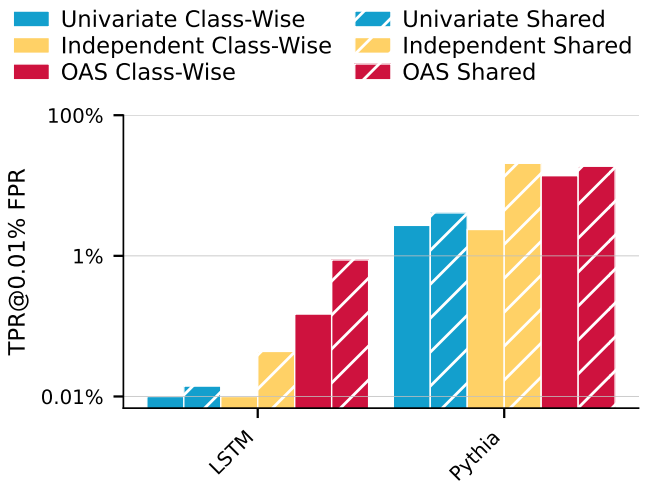


Figure 4: **The Univariate Class-Wise approach (Naive LiRA) results in a weaker attack.** TPR@0.01% FPR for various attacks using 10,000 average-case canaries with 64 shadow models. The Shared version consistently achieves better performance. Refer to Table 2, in Appendix F, for MIA performance with worst-case canaries.

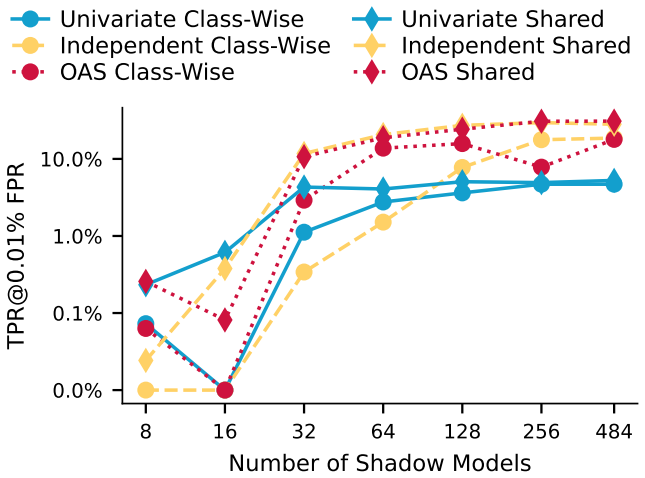


Figure 5: **The Univariate Shared attack is slightly better for small numbers of shadow models (< 16), but it is significantly worse for large numbers of shadow models.** The reported TPR@0.01% FPR results illustrate the performance of various attacks as the number of shadow models increases, based on experiments conducted with Pythia 1b and 10,000 average-case canaries.

instances of memorization with high confidence, aligning with standard practices in privacy auditing (see [5, 18] for a detailed discussion).

4.2. Comparison of Different Attacks

We first present an overview of LiRA adaptations using the different covariance estimators and a class-wise vs. shared covariance. Concretely, we compare all six attack variants using 64 shadow models with average-case canaries in Figure 4. Additionally, in Appendix F, Table 2 shows the results for the worst-case canaries and the image domain. Moreover, Appendix E highlights the Receiver Operating Characteristic (ROC) curve for the LSTM and Pythia average-case canaries, and shows that in these settings, the OAS systematically matches or beats the Univariate and Independent approaches across the whole curve.

First, compared to a naive baseline ("Univariate Class-Wise"), adaptations for sequence models seem to consistently uncover an order of magnitude more memorization in most cases. In particular, assuming homoscedasticity ("Shared") alone can significantly increase the TPR at a low FPR and never hurts. When considering worst-case canary, the models memorize significantly more, and in particular, the Independent Shared approach has a TPR @ 0.01% greater than 95% for both LSTM and Pythia 1b. In Appendix G, we examine various length reduction strategies and show that discarding certain tokens can enhance performance. For example, Figure 13 illustrates that the "Min" strategy—retaining only the two tokens with the smallest loss—yields a higher true positive rate (TPR).

The optimal attack method and length reduction strategy vary depending on the specific setting. In particular, it is important to analyze the covariance matrix to understand which prior information is useful in that specific setting. For instance, if the real covariance matrix is close to a diagonal matrix, and there is limited interaction between the tokens, then using the Independent approach leads closer to the real covariance matrix with a smaller number of shadow models than using the OAS method. Vice versa, if the real covariance matrix has many interactions the OAS approach becomes a more suitable choice.

4.3. Impact of the Number of Shadow Models

The number of shadow models plays a crucial role in calibrating the attack, as it directly influences membership inference performance. To better understand this relationship, we analyze how attack effectiveness changes as a function of the number of shadow models, using a setting where we have access to a larger pool of 484 shadow models. As illustrated in Figure 5, increasing the number of shadow models improves attack performance by uncovering more memorization, leading to higher TPR at low FPR. However, we observe key differences across methods. The univariate attacks plateau early, peaking at 32 shadow models, while more expressive estimators continue improving with additional models. Notably, for small numbers of shadow models (≤ 16), the Univariate Shared attack slightly outperforms others, suggesting that a simpler approach yields higher TPR in this setting. Additionally, shared covariance estimators consistently achieve comparable or superior TPRs with fewer shadow models than the class-wise approach, making them a more efficient choice when computational resources are limited.

5. Conclusion

In this work, we question the assumptions of standard membership inference attacks in the context of large sequence models. Based on the resulting insights, we introduce adaptations for existing membership inference attacks. Our results demonstrate that leveraging correlations between per-token losses, rather than relying solely on the average loss, significantly boosts attack performance. Most importantly, our adaptations enable strong audits with only few shadow models—a crucial requirement given the computational cost of training large sequence models. We hence provide an important stepping stone towards reliable memorization audits for large sequence models.

References

[1] F. Tramèr, G. Kamath, and N. Carlini, "Position: Considerations for differentially private learning with large-scale public pretraining," in *Forty-first International Conference on Machine Learning*, 2022.

[2] H. Brown, K. Lee, F. Mireshghallah, R. Shokri, and F. Tramèr, "What does it mean for a language model to preserve privacy?" in *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, 2022, pp. 2280–2292.

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

[4] J. Zhang, D. Das, G. Kamath, and F. Tramèr, "Membership inference attacks cannot prove that a model was trained on your data," *ArXiv preprint*, vol. abs/2409.19798, 2024. [Online]. Available: https://arxiv.org/abs/2409.19798

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

[6] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen *et al.*, "Palm 2 technical report," *ArXiv preprint*, vol. abs/2305.10403, 2023. [Online]. Available: https://arxiv.org/abs/2305.10403

[7] M. Reid, N. Savinov, D. Teplyashin, D. Lepikhin, T. Lillicrap, J.-b. Alayrac, R. Soricut, A. Lazaridou, O. Firat, J. Schrittwieser *et al.*, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *ArXiv preprint*, vol. abs/2403.05530, 2024. [Online]. Available: https://arxiv.org/abs/2403.05530

[8] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *ArXiv preprint*, vol. abs/2303.08774, 2023. [Online]. Available: https://arxiv.org/abs/2303.08774

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

[10] N. Carlini, M. Jagielski, C. Zhang, N. Papernot, A. Terzis, and F. Tramer, "The privacy onion effect: Memorization is relative," *Advances in Neural Information Processing Systems*, vol. 35, pp. 13 263–13 276, 2022.

[11] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramèr, and C. Zhang, "Quantifying memorization across neural language models," in *The Eleventh International Conference on Learning Representations*. OpenReview, 2023.

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

[13] M. Nasr, N. Carlini, J. Hayase, M. Jagielski, A. F. Cooper, D. Ippolito, C. A. Choquette-Choo, E. Wallace, F. Tramèr, and K. Lee, "Scalable extraction of training data from (production) language models," *ArXiv preprint*, vol. abs/2311.17035, 2023. [Online]. Available: https://arxiv.org/abs/2311.17035

[14] F. Pinto, N. Rauschmayr, F. Tramèr, P. Torr, and F. Tombari, "Extracting training data from document-based vqa models," in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML'24. JMLR.org, 2024.

[15] G. Somepalli, V. Singla, M. Goldblum, J. Geiping, and T. Goldstein, "Diffusion art or digital forgery? investigating data replication in diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6048–6058.

[16] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramèr, B. Balle, D. Ippolito, and E. Wallace, "Extracting training data from diffusion models," in *32nd USENIX Security Symposium (USENIX Security 23)*. Anaheim, CA: USENIX Association, 2023, pp. 5253–5270. [Online]. Available: https://www.usenix.org/conference/usenixsecurity23/presentation/carlini

[17] M. Aerni, J. Rando, E. Debenedetti, N. Carlini, D. Ippolito, and F. Tramèr, "Measuring non-adversarial reproduction of training data in large language models," *arXiv preprint arXiv:2411.10242*, 2024.

[18] M. Aerni, J. Zhang, and F. Tramèr, "Evaluations of machine learning privacy defenses are misleading," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2024, pp. 1271–1284.

[19] V. Feldman, "Does learning require memorization? a short tale about a long tail," in *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, 2020, pp. 954–959.

[20] W. Wang, M. A. Kaleem, A. Dziedzic, M. Backes, N. Papernot, and F. Boenisch, "Memorization in self-supervised learning improves downstream generalization," in *The Twelfth International Conference on*

Learning Representations, 2024.

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

[22] B. Jayaraman, L. Wang, K. Knipmeyer, Q. Gu, and D. Evans, "Revisiting membership inference under realistic assumptions," *Proceedings on Privacy Enhancing Technologies*, 2021.

[23] G. Ho, A. Sharma, M. Javed, V. Paxson, and D. Wagner, "Detecting credential spearphishing in enterprise settings," in *26th USENIX security symposium (USENIX security 17)*, 2017, pp. 469–485.

[24] T. Steinke and J. Ullman, "The pitfalls of average-case differential privacy," DifferentialPrivacy.org, 2020, https://differentialprivacy.org/average-case-dp/.

[25] T. Steinke, M. Nasr, and M. Jagielski, "Privacy auditing with one (1) training run," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[26] W. Shi, A. Ajith, M. Xia, Y. Huang, D. Liu, T. Blevins, D. Chen, and L. Zettlemoyer, "Detecting pretraining data from large language models," 2023.

[27] J. Zhang, J. Sun, E. Yeats, Y. Ouyang, M. Kuo, J. Zhang, H. Yang, and H. Li, "Min-k%++: Improved baseline for detecting pre-training data from large language models," *arXiv preprint arXiv:2404.02936*, 2024.

[28] J. Mattern, F. Mireshghallah, Z. Jin, B. Schoelkopf, M. Sachan, and T. Berg-Kirkpatrick, "Membership inference attacks against language models via neighbourhood comparison," in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023, pp. 11 330–11 343.

[29] A. Sablayrolles, M. Douze, C. Schmid, Y. Ollivier, and H. Jégou, "White-box vs black-box: Bayes optimal strategies for membership inference," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 5558–5567. [Online]. Available: http://proceedings.mlr.press/v97/sablayrolles19a.html

[30] L. Watson, C. Guo, G. Cormode, and A. Sablayrolles, "On the importance of difficulty calibration in membership inference attacks," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. [Online]. Available: https://openreview.net/forum?id=3eIrli0TwQ

[31] F. Tramèr, R. Shokri, A. San Joaquin, H. Le, M. Jagielski, S. Hong, and N. Carlini, "Truth serum: Poisoning machine learning models to reveal their secrets," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 2779–2792.

[32] S. Zarifzadeh, P. Liu, and R. Shokri, "Low-cost high-power membership inference attacks," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, Eds., vol. 235. PMLR, 21–27 Jul 2024, pp. 58 244–58 282. [Online]. Available: https://proceedings.mlr.press/v235/zarifzadeh24a.html

[33] J. Neyman and E. S. Pearson, "Ix. on the problem of the most efficient tests of statistical hypotheses," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 231, no. 694-706, pp. 289–337, 1933.

[34] H. Larochelle and I. Murray, "The neural autoregressive distribution estimator," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 29–37.

[35] A. El-Nouby, M. Klein, S. Zhai, M. Á. Bautista, V. Shankar, A. T. Toshev, J. M. Susskind, and A. Joulin, "Scalable pre-training of large autoregressive image models," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=c92KDfEZTg

[36] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, "PixelCNN++: Improving the pixelCNN with discretized logistic mixture likelihood and other modifications," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=BJrFC6ceg

[37] G. E. Box and G. M. Jenkins, "Some recent advances in forecasting and control," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 17, no. 2, pp. 91–109, 1968.

[38] J.-E. Shin, A. J. Riesselman, A. W. Kollasch, C. McMahon, E. Simon, C. Sander, A. Manglik, A. C. Kruse, and D. S. Marks, "Protein design and variant prediction using autoregressive generative models," *Nature communications*, vol. 12, no. 1, p. 2403, 2021.

[39] S. Hochreiter, "Long short-term memory," *Neural Computation MIT-Press*, 1997.

[40] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[41] S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff *et al.*, "Pythia: A suite for analyzing large language models across training and scaling," in *International Conference on Machine Learning*. PMLR, 2023, pp. 2397–2430.

[42] Y. Chen, A. Wiesel, Y. C. Eldar, and A. O. Hero, "Shrinkage algorithms for mmse covariance estimation," *IEEE transactions on signal processing*, vol. 58, no. 10, pp. 5016–5029, 2010.

[43] J. Ye, A. Maddi, S. K. Murakonda, V. Bindschaedler, and R. Shokri, "Enhanced membership inference attacks against machine learning models," in *Proceedings of the 2022 ACM SIGSAC Conference on Com-*

puter and Communications Security, 2022, pp. 3093–3106.

[44] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth *et al.*, "Gemini: a family of highly capable multimodal models," *ArXiv preprint*, vol. abs/2312.11805, 2023. [Online]. Available: https://arxiv.org/abs/2312.11805

[45] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston, "Personalizing dialogue agents: I have a dog, do you have pets too?" in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, I. Gurevych and Y. Miyao, Eds. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 2204–2213. [Online]. Available: https://aclanthology.org/P18-1205/

[46] I. Loshchilov, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

Appendix

A Figure Setup

Here, we describe the setup for the most relevant figures.

Figure 1. The two groups of bars refer to the LSTM, and the Pythia 1b, which are the two language settings analyzed, using average-case canaries. As described in Section 4.1, we used 10,000 samples and 64 shadow models. The "Naive Approach", which corresponds to LiRA as-is, where membership guesses only depend on a sample's loss, corresponds to the Univariate Class-Wise approach, while "Our Approach" corresponds to OAS Shared.

Figure 2 and Figure 3. The figures focus on the case with Pythia 1b and average-case canaries because we have access to a larger number of shadow models. To obtain the most reliable estimation of the full covariance matrix, we use all 484 shadow models and apply the MLE, which serves as gold standard. Our analysis is based on 1,000 samples from the average-case canaries. In Figure 2, the y-axis represents the covariance approximation error, measured as the Frobenius norm between the gold standard estimate and the covariance matrix obtained using different methods with varying numbers of shadow models. We choose the Frobenius norm as it is one of the most commonly used norms for matrices. For Figure 3, the y-axis represents the relative covariance error, which quantifies the relative difference between the covariance approximation error of the shared estimated covariance matrix and that of the class-wise estimated covariance matrix (class-wise error minus shared error, normalized by the shared error). A higher relative covariance error indicates that the shared estimation is better (a smaller covariance approximation error) than the class-wise one.

B Additional Covariance Estimations

Figure 6a and Figure 6b present the corresponding results using the OUT case samples, analogous to Figure 2 and Figure 3, respectively. The OUT case (distribution of the nonmembers' losses) shows the same trends as the IN case (distribution of the members' losses).

C Case Study: Autoregressive Image Generator

We also consider a case study using Pixel-CNN++ [36], an autoregressive image generator.

Experimental Settings. We train 64 instances of Pixel-CNN++ [36] on CIFAR-10. Due to the computational cost, the models are trained on a smaller number of epochs, compared to the original implementation which trained the model for 5000 epochs, the model is slightly under-trained leading to poor membership inference performance. Therefore, this shows an extremely challenging setting. We use samples from the test set as canaries. Compared to the language setting, where the sequence length is 128, in the image case the sequence length depends on the size of the image which is 32 x 32, and therefore the sequence length is 1024. This further increases the computational cost and the complexity of the task.

Attack Evaluation. Figure 7 shows the performance of the MIAs with an Autoregressive Image Generator. Overall, the performance is quite poor. For this reason, we highlight the MIA performance using TPR @ 1% FPR instead of TPR @ 0.01 FPR. In Appendix F, Table 2 confirms that the performance for 0.1% and 0.01% is close to random guessing. OAS is the only attack that performs better than random guessing for TPR @ 1% FPR. The reason is that there is a stronger interaction between tokens than in the language domain. To illustrate this, Figure 8 shows the covariance matrix of the per-token loss distribution of the members. We observe that the structure is more complex and is different from the Independent one, as the covariance matrix has nonzero values outside the diagonal.

D Choice of the hyperparameters

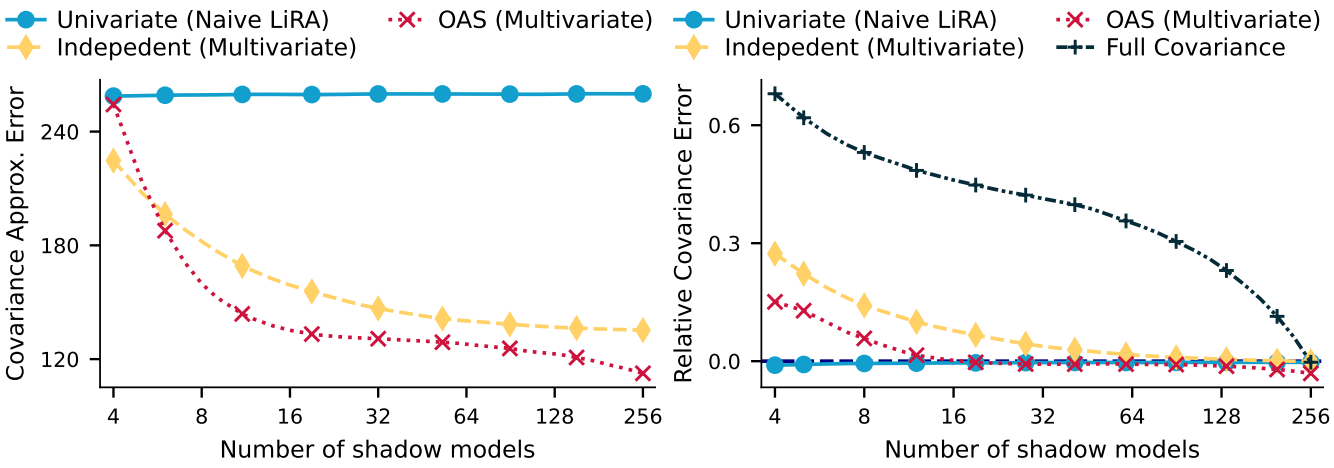
Table 1 shows the selected hyperparameters for each setting. In all the settings, we trained the models using AdamW [46].

E ROC Curve

Figure 9 and Figure 10 show the log-scale Receiver Operating Characteristic (ROC) curve of the success rates of the different types of attacks for the LSTM and Pythia Average. In particular, Figure 9 shows the ROC curve using the two covariances (Class-Wise), while the Figure 10 shows the results with a single shared covariance matrix (Shared). In both cases, the results are for the average-case canaries. We see that the OAS approach has a higher TPR compared to the other in particular, for low FPR, where privacy auditing is more important.

F Detailed Results

Table 2 shows the results for each setting. First, we observe that the sequence-aware attacks (Independent and



(a) The Covariance Approximation Error, measured as the Frobenius norm between the covariance matrix estimated using 484 shadow models and the full covariance MLE, is compared to the covariance matrix estimated with a given method and varying numbers of shadow models. As the number of shadow models increases, we analyze this error using 1,000 average-case canaries from the OUT case of Pythia 1b with a sequence length of 128.

(b) The Relative Covariance Error is measured as the relative difference between the Covariance Approximation Error of the Shared estimated covariance matrix and the Class-Wise one. As the number of shadow models increases, we analyze this error using 1,000 average-case canaries from the IN case of Pythia 1b with a sequence length of 128.

Figure 6: **The OUT case follows the same trends as the IN case.**

Setting	Epochs	Learning Rate	Weight decay	Batch Size	Hidden Dimension	Sequence Length
LSTM	10	10^{-3}	0.0	64	192	128
Pythia 1b	7	10^{-4}	10^{-4}	16	/	128
Pixel-CNN++	50	10^{-3}	0.0	192	192	1024

TABLE 1: **The selected hyperparameters for each setting**

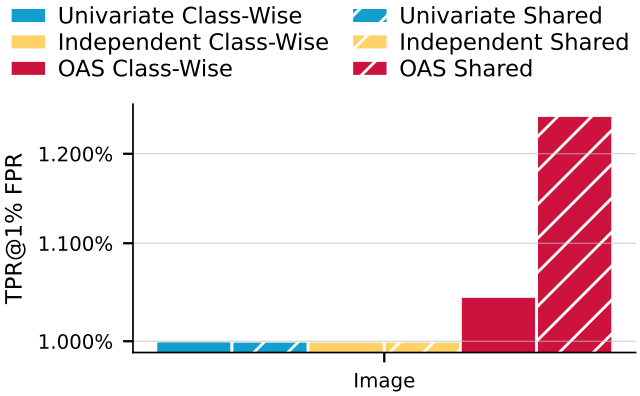


Figure 7: **OAS is the only approach that performs better than random guessing.** The TPR @ 1% FPR using PixelCNN++ for average-case canaries.

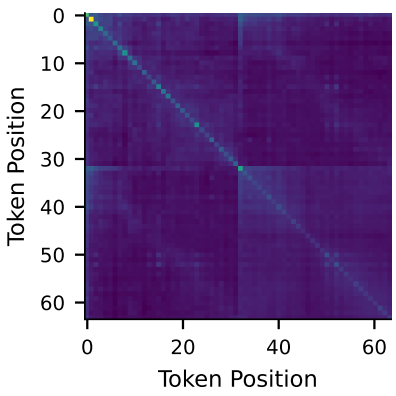


Figure 8: **The covariance structure is not diagonal.** The covariance matrix of the OUT case is estimated using the full covariance MLE on the distribution of the first 64 tokens with PixelCNN++ using 64 shadow models.

OAS) obtain higher TPRs across all the settings. For the image case, all the attacks have a close to random guessing performance due to the complexity of the task.

G Comparison of Length Reduction Strategies

We further evaluate length reduction strategies that aim to select a representative subsequence for membership inference. Grouping corresponds to chunking the tokens in groups and computing the average of each group, therefore, when the sequence length is 1, this corresponds to Naive LiRA (Univariate Approach), and using all the 128 tokens corresponds to the standard baseline (either Independent or OAS). Figures 11 to 14 show the TPR @ 0.01% FPR for

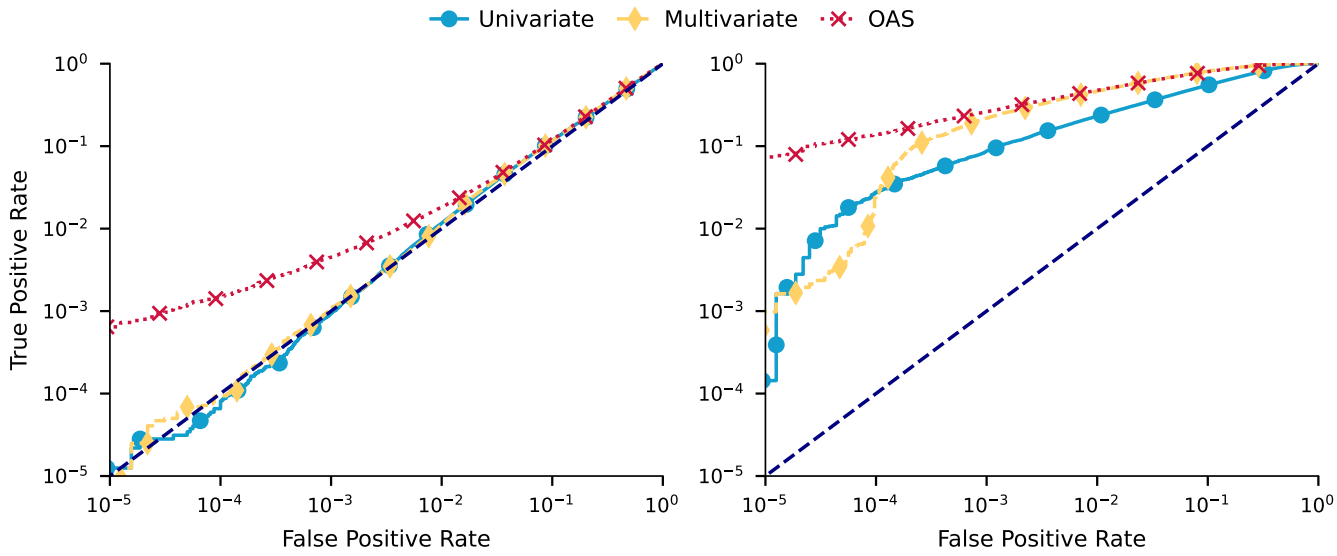


Figure 9: **Using the Naive Approach (Univariate) gives suboptimal results.** Comparing the true positive rate vs. false positive rate for different ways to model the LiRA IN and OUT distributions in the class-wise case. On the left, we use an LSTM, and on the right Pythia 1b using average-case canaries.

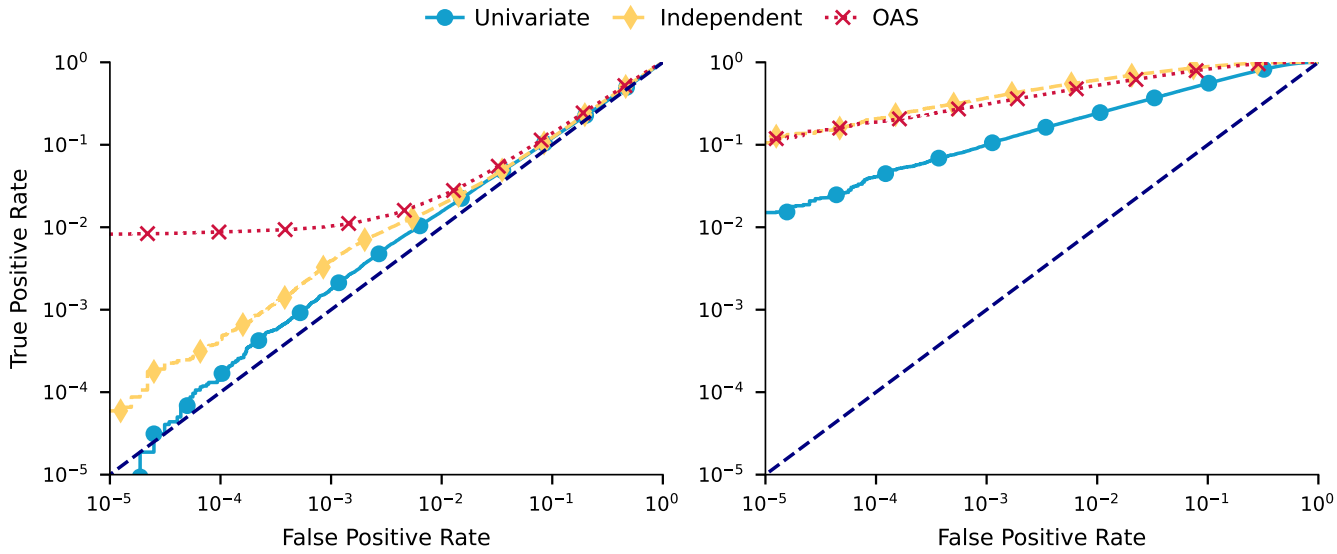


Figure 10: **Using Naive Approach (Univariate) gives suboptimal results, also when considering a shared covariance matrix.** Comparing the true positive rate vs. false positive rate for different ways to model the LiRA IN and OUT distributions in the shared case. On the left, we use an LSTM, and on the right Pythia 1b using average-case canaries.

different length reduction strategies. For instance, Figure 11, which represents Pythia 1b with the average-case canaries, shows that using all the tokens is beneficial, however, it is not always the case. For instance, when evaluating the LSTM with average-case setting, Figure 13 shows that Min with a reduced sequence length of 2 gives the best MIA.

MIA	Pythia Average		Pythia Worst		LSTM Average		LSTM Worst		Image	
	0.1%	0.01%	0.1%	0.01%	0.1%	0.01%	0.1%	0.01%	0.1%	0.01%
Univariate Class-Wise	8.69	2.72	95.04	89.66	0.10	0.01	92.51	4.33	0.10	0.01
Univariate Shared	10.01	4.09	94.91	91.26	0.18	0.01	95.82	88.25	0.10	0.01
Independent Class-Wise	21.78	2.37	99.58	99.18	0.11	0.01	91.52	0.02	0.10	**0.02**
Independent Shared	**36.98**	**20.72**	99.67	99.30	0.39	0.04	**98.75**	**95.00**	0.10	0.01
OAS Class-Wise	26.26	13.81	**99.78**	**99.57**	0.45	0.15	23.19	0.03	**0.11**	0.01
OAS Shared	30.95	18.81	99.67	99.37	**1.03**	**0.87**	70.74	30.72	0.10	0.01

TABLE 2: **The univariate attacks always under-perform the sequence-aware attacks.** The TPR@{0.1,0.01}% FPR for different types of attacks and settings. All the scores are computed using 10,000 canaries and 64 shadow models using leave-one-out.

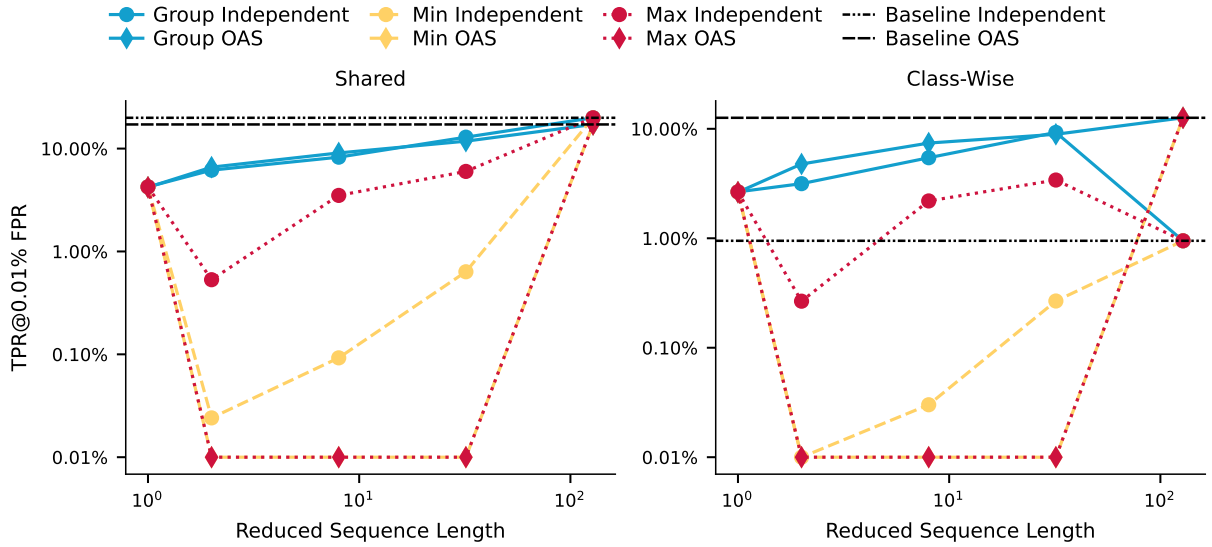


Figure 11: **Pythia - Average-case canaries.** TPR @ 0.01% FPR for different length reduction strategies on Pythia 1b with the average-case canaries. The black lines represent the baselines, where no grouping strategy is applied, and all the tokens in the sequence are used. The dashed lines represent the OAS case, while the solid lines represent the Independent case.

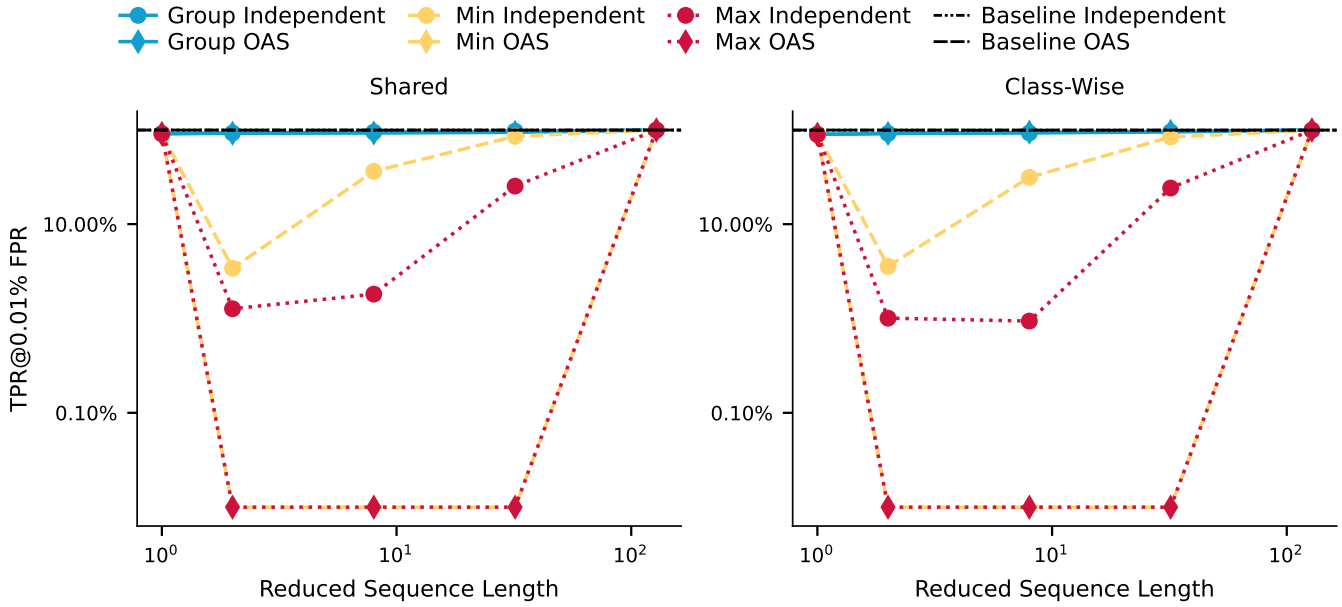


Figure 12: **Pythia - Worst-case canaries.** TPR @ 0.01% FPR for different length reduction strategies on Pythia 1b with the worst-case canaries. The black lines represent the baselines, where no grouping strategy is applied, and all the tokens in the sequence are used. The dashed lines represent the OAS case, while the solid lines represent the Independent case.

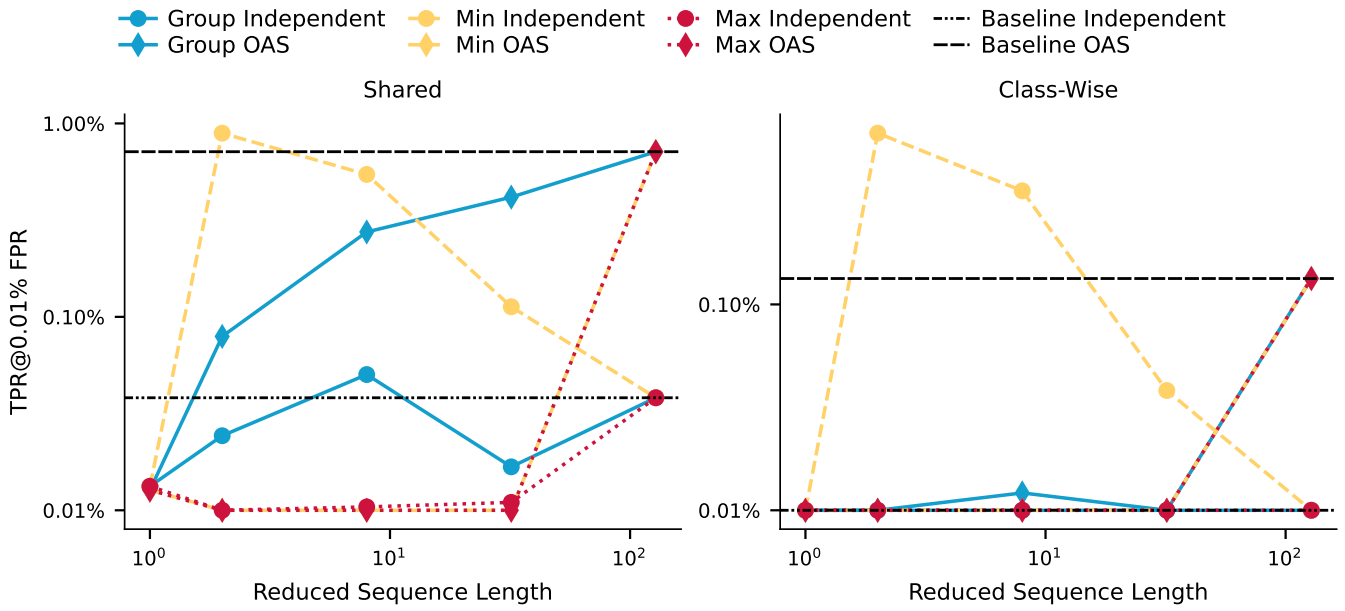


Figure 13: **LSTM - Average-case canaries.** TPR @ 0.01% FPR for different length reduction strategies on LSTM with the average-case canaries. The black lines represent the baselines, where no grouping strategy is applied, and all the tokens in the sequence are used. The dashed lines represent the OAS case, while the solid lines represent the Independent case.

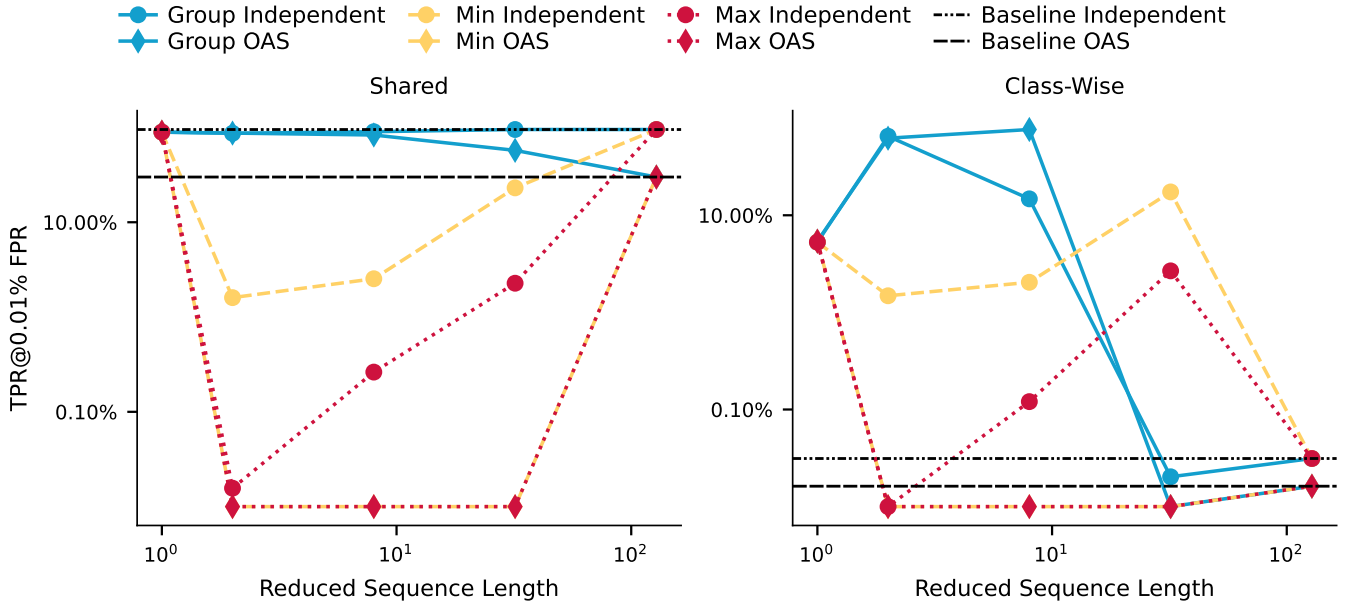


Figure 14: **LSTM - Worst-case canaries.** TPR @ 0.01% FPR for different length reduction strategies on LSTM with the worst-case canaries. The black lines represent the baselines, where no grouping strategy is applied, and all the tokens in the sequence are used. The dashed lines represent the OAS case, while the solid lines represent the Independent case.