CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Semantically Aware Attacks on Text-based Models

An Extension of Context-aware and Neighbourhood Comparison-based Membership Inference Attacks

Master's thesis in Computer science and engineering

GABRIEL GLÄNTE

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2025

Master's thesis 2025

Semantically Aware Attacks on Text-based Models

An Extension of Context-aware and Neighbourhood Comparison-based Membership Inference Attacks

GABRIEL GLÄNTE

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2025

Semantically Aware Attacks on Text-based Models
An Extension of Context-aware and Neighbourhood Comparison-based Membership Inference Attacks
GABRIEL GLÄNTE

Supervisor: Anton Matsson, Computer Science and Engineering
Advisor: Johan Östman, AI Sweden
Advisor: Fazeleh Hoseini, AI Sweden
Examiner: Fredrik Johansson, Computer Science and Engineering

Master's Thesis 2025
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2025

Semantically Aware Attacks on Text-based Models
An Extension of Context-aware and Neighbourhood Comparison-based Membership Inference Attacks
GABRIEL GLÄNTE
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Training deep-learning models requires large amounts of data. When this data is sensitive, e.g., containing personal information, it is important to ensure that no sensitive information can be extracted from the trained models. In a membership inference attack (MIA), an adversary is expected to have access to a trained model θ and a data sample d, sampled from the same distribution as the unknown training data. The objective of the adversary is to construct an algorithm $A(\theta, d) \to \{0, 1\}$, where the binary output guesses if d was part of the unknown training data or not. It is commonly assumed that the attacker can access loss values from θ for different prompts; such loss-based signals are crucial for membership checks, even under black-box conditions.

For text, the notion of membership is not clear-cut: distinct strings can share the same semantics. Many MIAs therefore fail when they only test exact strings. Recent work reports near-random performance across models and domains (15). This suggests the need to incorporate semantics, i.e., to probe a text together with semantic neighbours that preserve meaning under small, context-appropriate edits. This thesis explores and strengthens such attacks and evaluates them with the standard metrics area under the ROC curve (AUC) and true positive rate at low false-positive rates (TPR@1%FPR).

Building on the context-aware membership inference attack (CAMIA) which uses per-token loss sequences rather than a single average loss to construct signals for membership inference (11), the contributions of this thesis are: (i) a custom re-implementation of CAMIA, (ii) integrating a neighbourhood comparison signal that perturbs a text with its semantic neighbours (16), and (iii) novel signals designed to improve loss-informed neighbour generation. Experiments on Pythia-deduped and GPT-Neo models across six subsets of The Pile (19) (streamed via the MIMIR repository (15)) show that these semantics-aware extensions often increase true positive rates at low false positive rates while keeping AUC stable. Overall, modest, loss-guided semantic edits make MIAs more effective for text under realistic black-box conditions.

Keywords: membership inference attack, large language model, privacy, semantic perturbation, neighbourhood comparison

Acknowledgements

Thank you Chalmers University of Technology, for an excellent education and for the friendships formed over the past five years.

Many thanks to Johan Östman, Fazeleh Hoseini, and all colleagues at AI Sweden for an exciting and collaborative research environment. I also thank Anton Matsson and Fredrik Johansson for their supervision, clear guidance, and timely feedback throughout the project.

I am deeply grateful to my family, friends, and my fiancée Luzia for your patience, encouragement, and unwavering support. Thank you so much for helping me become an engineer and fulfilling this dream.

Gabriel Gläntе, Gothenburg, 2025-09-11

Contents

List of Figures

List of Tables

1

Introduction

Training deep learning models requires large amounts of data. When these data are sensitive, for example containing personal information, it is important to ensure that sensitive information cannot be extracted from the trained models. Lately, adversarial attempts to extract training data have grown in interest. In this context, an "attack" refers to any method an adversary might use to glean private or sensitive data from a trained model. Two prominent examples are membership inference attacks, which attempt to guess if a given data point was present in the training data, and reconstruction attacks, also called model-inversion attacks, which attempt to recreate training data by interacting with the trained model.

Although such attacks are relevant for any data modality, perhaps the most pressing issue pertains to text-based models, where copyright concerns have recently appeared in the media, such as the lawsuit against OpenAI (8). In light of this, there is also a pressing need for content creators to confidently test whether or not their output has been included and used in the training of commercial models. Membership inference attacks offer a promising venue for such an assessment (5).

1.1 Context

AI Sweden is currently leading a project within adversarial information extraction against trained machine learning models. The project is called LeakPro and is a collaboration that includes RISE, Sahlgrenska, Region Halland, Astra Zeneca, Syndata, and Scaleout. The main goal of LeakPro is to create an open-source tool to stress test trained machine learning models to understand the risk of leaking sensitive information from training data. Multiple categories of stakeholders would benefit from such a tool. Currently, LeakPro supports image, tabular, and graph data, but the platform is designed to be agnostic to data modalities.

In a membership inference attack, an adversary is expected to have access to a trained model θ and a data sample d, sampled from the same distribution as the unknown training data (or at least having support on that distribution). The objective of the adversary is to construct an algorithm $A(\theta, d) \to \{0, 1\}$, where the binary output guesses if d was part of the unknown training data or not. Since many large language models offer next-token probabilities or log-probabilities via standard API endpoints, it is commonly assumed that the attacker can access some form of loss for each query;

such loss-based signals are crucial for membership checks, even under black-box conditions.

For text, the definition of a data point being part of a dataset is not clear-cut. For example, many membership inference attacks (5; 6; 3) attempt to detect if a specific sentence was part of the dataset. However, this does not account for different text snippets that share equivalent semantics (texts with the same meaning but different words). Moreover, many user scenarios involve longer text fragments, where the possibility of paraphrasing or reorganizing content becomes more relevant. Recognizing such equivalences is crucial as longer texts might be reworded while retaining the same underlying meaning.

1.1.1 Motivation for Semantically Aware Attacks

MIMIR (a unified repository for evaluating membership inference attacks on language models (15)), presented at the 2024 Conference on Language Modeling, evaluated membership inference attacks (MIAs) on a wide range of pre-trained large language models (LLMs) and found that "the performance across most MIAs and target domains is near-random." This stark finding underscores an urgent need to design better MIAs for pre-trained LLMs. (11)

Recent work (12; 13) suggests that incorporating semantics or searching not only for an exact data point but also in its immediate vicinity can greatly improve MIA effectiveness. This thesis aims to explore and strengthen such attacks, measuring performance with the standard metrics area under the ROC curve (AUC) and true positive rates at low false positive rates (TPR@1%FPR).

1.1.2 Two Uses of Semantics

In designing membership inference attacks that incorporate semantics, it is crucial to define the term and assess why one would want such attacks. Two potential approaches are suggested below.

An attack that incorporates semantics could, in theory, discover whether certain information is memorized, irrespective of the exact format or structure. For example, consider an attacker testing whether a model has been trained on data specifying or indicating the location of a subject. By testing "John Doe's location is x" for different values of x and observing, e.g., which variation produces the lowest cross-entropy loss on the model, the attacker may infer the true location, even if the original dataset did not contain an exact matching string. Such an approach could reveal sensitive details that are memorized in any rephrased form. Such an attack would be powerful, since only the semantics of the prompt would be important, not the exact structure or syntax. In other words, sufficiently sophisticated semantics-based attacks could in theory systematically brute-force or guess critical content.

Another scenario is to verify the membership status of a known data point, even if it has been slightly edited. For instance, if a copyrighted text was used in training, the model's lower loss on minor variants (with small Hamming distance, extra filler

words, etc.) could confirm or strongly indicate memorization. It could thus be possible to show that the original copyrighted snippet remains memorized by the model, despite trivial textual alterations.

The fundamental difference is that the first approach tries to capture information regardless of format or modality, while the second focuses on an exact data point, possibly with small edits. Attacks in the first category are arguably more complex; if they succeed in finding certain memorized information, it also implies the ability to detect at least one data point conveying that information (otherwise, it could not find the information in the first place). However, an approach restricted to verifying a single snippet might fail to detect more drastic paraphrases that convey the same content. Given the current state of research and the limited time available for this project, it is more tractable to develop an attack aligned with the second type: detecting membership of a known data point and its closely related variants. In fact, existing work (12; 13) has focused on testing lightly modified or paraphrased text to see if membership signals remain. This serves as a foundation for potential future attacks that could handle broader forms of semantic equivalences.

Based on this reasoning, one goal in this thesis is to work with text snippets with sufficiently high semantic similarity, to enable membership detection of original samples. In practice, this translates to minimal edits rather than complete rephrasings, although more extensive transformations are desirable for future research. Previous work (13) already notes that performance quickly degrades with increasingly comprehensive alterations.

1.1.3 Existing Approaches and Feasibility

Note that a trivial solution to "detecting" membership of a data point that is present in the training data of a model is simply to prompt the model directly. This is particularly the case if the information the data point carries is understood by the model and not just memorized. However, there may be guardrails in place, such as instructions for the model to not share sensitive information (21), or the model might occasionally hallucinate (20), such that concrete statements may not be trusted without proof.

Therefore, a more systematic approach is warranted, such as a membership inference attack to detect training data. Starting with minimally edited samples to maintain semantic equivalence is a pragmatic first step toward a fully information-level MIA. However, this limits the extent to which semantics as a concept can be experimented with. An important consideration for the research to be conducted for this thesis is therefore to also consider other recent MIAs - not only focused on semantics but also on related topics. Section 2.2 summarizes multiple recent such papers (11; 16; 13; 12) covering novel MIAs focused on context-awareness, neighbourhood comparison, semantics and range queries. The approaches in (16; 13) (The Neighbourhood Comparison Attack (NCA) and Semantic MIA (SMIA)) show that semantics-preserving perturbation of tokens improve performance, but the choice of which tokens or words in a sentence to be perturbed to semantically equivalent or similar variants is random or arbitrary (see Sections 2.2.3.2 and 2.2.4). A hypothesis in this thesis is

that the selection of word or token to perturb affects performance and that there might exist tractable strategies to identify suitable indices in a text. The recently proposed Context-Aware Membership Inference Attack (CAMIA) (11) introduces an interesting MIA where cross-entropy loss is computed per token (instead of for the entire text at once), taking the context of each word into account. Calculating the loss of each token generates informative sequences that form the basis for various signals useful for membership inference (Section 2.2.2). Such an approach could be a suitable target for experiments on token-wise semantic perturbations. Given that the cross-entropy loss per token is already computed in CAMIA, perturbing tokens to semantically equivalent or similar tokens based on their loss could constitute a feasible starting point for further improvements. However, a limitation is that there is no public code for CAMIA available, which requires a custom implementation to allow experiments to be possible.

1.2 Research Questions

This thesis proposes to integrate insights from recent membership inference attacks into a new combined architecture, and to develop them further in order to increase performance. The research questions follow below.

RQ1: For semantics-preserving text edits in membership inference attacks, can targeted and loss-guided token perturbations improve performance over masking random or arbitrary words?

RQ2: Can a unified attack that combines and develops recent MIAs improve current state-of-the-art results?

1.3 Contributions

This thesis makes three concrete contributions:

1. a custom, open-source implementation of the context-aware membership inference attack CAMIA (11) [1];
2. improving CAMIA results through integration of the Neighbourhood Comparison Attack (16) as an additional signal family (Section 3.3); and
3. additional performance improvements through the design and implementation of novel next-token loss sequence-based signals (Section 3.4).

[1] Code: `https://github.com/gabrielglante/LeakPro`

2

Theory

The purpose of the following chapter is to provide a theoretical foundation for answering the research questions. The chapter contains definitions and descriptions of two essential and standard metrics to measure membership inference attack performance: area under the ROC curve (AUC) and true positive rates at low false positive rates (TPR@1%FPR). It also covers summaries of recent approaches relevant to assess the influence of tokens perturbed to semantically equivalent variants on membership inference. Other covered approaches are related to semantics, such as context-dependence and range queries. Four papers are discussed, out of which two have been replicated and incorporated into experiments conducted in this thesis: Context-Aware Membership Inference Attacks against Pre-trained Large Language Models (CAMIA) (11) and Membership Inference Attacks against Language Models via Neighbourhood Comparison (16). The other two are relevant for future extensions and further development: Semantic Membership Inference Attack against Large Language Models (13) and Range Membership Inference Attacks (12). For correctness, the notation from each paper is preserved in its respective summary. The differences in notation reflect that the research area has not been standardized, and that membership inference attacks are part of a growing research field. Brief summaries of logistic regression and principal component analysis are also included in this chapter, as these techniques are employed in CAMIA.

2.1 Key metrics

2.1.1 Area Under the ROC Curve (AUC)

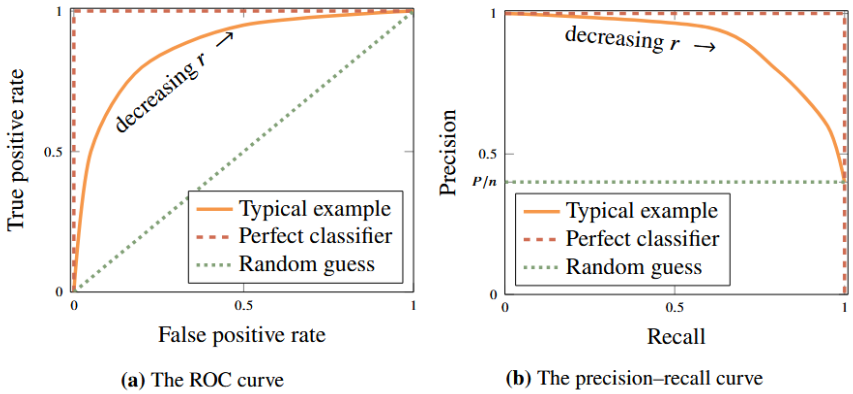
A natural starting point for assessing the performance of membership inference attacks is to enumerate the number of correct and incorrect predictions of data points belonging and not belonging to a validation dataset. Such data points can be defined as members and non-members, respectively. As described in (25), these predictions can be categorized as true and false positives and negatives in a confusion matrix:

	$y = 0$	$y = 1$	total
$\hat{y} = 0$	TN	FN	N^*
$\hat{y} = 1$	FP	TP	P^*
total	N	P	n

Here, comparing the counts of the true labels $y \in \{0, 1\}$ with the predicted $\hat{y} \in \{0, 1\}$, the numbers of true and false positives and negatives $\mathrm{TP}, \mathrm{FP}, \mathrm{FN}, \mathrm{TN}$ are recovered. The number of positive instances $P = \mathrm{TP} + \mathrm{FN}$, the number of negatives $N = \mathrm{TN} + \mathrm{FP}$, the number of predicted positives $P* = \mathrm{FP} + \mathrm{TP}$ and predicted negatives $N* = \mathrm{TN} + \mathrm{FN}$ are also obtained. Examples of commonly used metrics using these counts are

$$\text{True positive rate (TPR)} = \frac{\mathrm{TP}}{P}, \qquad \text{False positive rate (FPR)} = \frac{\mathrm{FP}}{N},$$

also known as recall (or sensitivity) and fall-out, respectively. Furthermore, many models output a real-valued score $s(x)$ on an input x. The higher the score, the stronger the evidence for the positive class. A binary decision is obtained by setting a threshold r and predicting $\hat{y} = \mathbb{1}[s(x) \geq r]$. Lowering r converts some negatives to positives, simultaneously increasing (or at least not decreasing) both TPR and FPR. The plot of TPR against FPR for every possible threshold traces the receiver–operating characteristic (ROC) curve. The curve runs from $(0, 0)$ (no positives predicted) to $(1, 1)$ (everything predicted positive). A perfect classifier touches the upper left corner, and collapses to the diagonal TPR = FPR for random guessing.



(a) The ROC curve **(b)** The precision–recall curve

Figure 2.1: The ROC and precision-recall curves (Figure 4.13 from (25))

A metric which captures the entire ROC graph is the area under the curve (AUC), which is equivalent to the probability that the classifier (such as a MIA) scores a data point from the positive class (such as a member) higher than one from the negative class (such as a non-member). A perfect classifier therefore has an AUC score of 1, and a classifier performing no better than random a score of 0.5. (25)

Formally, AUC can be calculated as follows. Let S_P and S_N be the scores assigned to randomly drawn positive and negative examples, F_P and F_N the cumulative distribution of the scores of the positive and negative classes, and $f_P(r)$ and $f_N(r)$ the corresponding probability density functions. Then:

$$F_P(r) = \Pr\Big(S_P < r\Big) \;=\; \int_{-\infty}^{r} f_P(s)\, ds, \tag{12.4}$$

$$F_N(r) = \Pr\Big(S_N < r\Big) \;=\; \int_{-\infty}^{r} f_N(s)\, ds. \tag{12.5}$$

The corresponding true-positive and false-positive rates at threshold r are

$$v(r) = 1 - F_P(r), \qquad u(r) = 1 - F_N(r).$$

These can now be used to define:

$$\text{AUC} \;=\; \int_0^1 v\Big(r^{-1}(u)\Big)\, du, \tag{12.6}$$

where $r^{-1}(u)$ is the inverse of $u(r)$. Finally, AUC is equivalent to the probability that the positive member receives a higher score than the negative member:

$$\begin{aligned}\text{AUC} &= \int_{\infty}^{-\infty} v(r)\,(-f_N(r))\, dr = \int_{-\infty}^{\infty} v(r)\, f_N(r)\, dr \\ &= \int_{-\infty}^{\infty} f_N(r)\,(1 - F_P(r))\, dr = \Pr\Big(S_P < S_N\Big).\end{aligned} \tag{12.7}$$

Since the ROC curve sweeps over the whole false-positive range, it is most informative when the class proportions are roughly balanced and the cost of FP and FN errors is comparable. In imbalanced or asymmetric scenarios where most data points belong to one class or the impact of false negatives are not equivalent to false positives, other metrics can be more useful. A de facto standard metric in the context of membership inference attacks is described in the next section.

2.1.2 True-Positive Rates at Low False-Positive Rates

Average case scores such as AUC show how an average sample behaves when sweeping the decision threshold across its entire range. However, membership inference attacks should arguably not be judged by their average performance. An adversary only needs to correctly expose a single or handful of training examples to breach confidentiality, whereas even a single false accusation can be unacceptable in practice. Carlini et al. (10) therefore argue that membership inference attacks should be evaluated by the true positive rate (TPR) that can be achieved while keeping the false positive rate (FPR) very small (e.g. 0.1% or below). The authors show that many attacks, with balanced accuracy scores much better than random guessing, perform worse than chance at low false positive rates. For example, the popular Loss threshold attack achieves 60% balanced accuracy on CIFAR-10, yet yields zero true positives at 0.1% FPR. The Loss attack is also shown to be a strong non-membership inference

attack, and practically useless for identifying members. These results call for better membership inference attacks and a general adoption of suitable metrics.

By fixing the false positive rate at, e.g., 1%, the TPR@1%FPR metric thus reports the true positive rate ($TP/(FN+TP)$) for classifying membership given that at most 1 in 100 non-members are incorrectly classified as members ($FP/(TN+FP) \leq 0.01$). Carlini et al. perform experiments with the FPR as low as 0.001% FPR, inspired by standards in computer security where false alarms are also costly. Recent research in the field (11; 12; 13; 15; 16) have adopted this metric, and can to an increasing extent be considered a standard.

2.2 Summaries of Recent Approaches

In recent literature, multiple new methods incorporate partial or full notions of semantics in their membership inference attacks. This section briefly summarizes each approach, highlighting the essential formulas and algorithmic steps. Note that each approach relies on some form of loss from the model outputs, which is typically accessible even in black-box scenarios through token-level log probabilities or similar statistics. Note that the notation used below is based on the notation in each summarized paper respectively. There are therefore slight differences in notation for each paper.

2.2.1 Previous Attacks

A previous baseline attack is the Loss-based attack (LOSS) (27), which predicts membership based on whether a large language model's loss over all tokens of a text is below a certain threshold. There also exist membership inference attacks tailored to pre-trained large language models utilizing the LOSS attack. Examples of such attacks, which are compared against in the experiments of more recent approaches such as CAMIA (summarized below), are the Min-K%, MinK%++, Reference-based, Neighbourhood and Zlib attacks. The Min-K% attack (26) is based on the hypothesis that non-member samples are more probable to contain a small number of outlier tokens with very high loss compared to members. The scoring function is based on the average loss of the K% of tokens with the lowest likelihoods. The Min-K%++ attack (29) is based on the same logic, but the loss score is normalized with the expectation and standard deviation of the next token's log probability over the vocabulary of the large language model given its prefix. The Reference-based attack (28) uses a membership scoring function which compares the cross-entropy loss of a model targeted for an attack against a reference model trained on same-distribution, but mostly disjoint data. The Zlib attack (28) normalizes the target model's loss of a sample with its zlib entropy (the number of bits of entropy when the sequence is compressed with zlib compression (28)). Out of these, the Neighbourhood Comparison Attack is summarized in greater detail later in this chapter, as it is incorporated in the attack architecture designed in this thesis. Note that the contents of the paper Context-Aware Membership Inference Attacks against Pre-trained Large Language Models (11) is described in greater detail than the

others, as the corresponding attack has been implemented as a foundation for the experiments conducted for this thesis.

2.2.2 Context-Aware Membership Inference Attack (CAMIA)

CAMIA (11) is a recently proposed attack framework designed specifically for black-box membership inference on pre-trained LLMs. The key insight behind CAMIA is that membership signals in LLMs often manifest at the token level - that is, through the sequence of next-token losses or probabilities across an entire text. Standard "loss-based" attacks typically aggregate model loss over the entire input text (e.g., compute a single mean loss), thus missing important token-by-token behaviors and ignoring how prefixes influence subsequent predictions. CAMIA overcomes these limitations by analyzing finer-grained signals that reflect contextualized memorization within the LLM.

The notation in (11) can be summarized and expressed as follows. If $\mathcal{M}$ is an auto-regressive LLM with vocabulary $\mathcal{V} = \{v_1, \ldots, v_{|\mathcal{V}|}\}$, the token sequence is $\mathbf{X} = [x_1, x_2, \ldots, x_T] \quad (x_t \in \mathcal{V})$ after tokenization of an input text. The prefix of length $t-1$ can be written as $\mathbf{x}_{<t} = [x_1, \ldots, x_{t-1}]$, and can receive a next-token loss sequence at every step t from the $\mathcal{M}$ output distribution $P(x_t \mid \mathbf{x}_{<t}; \mathcal{M})$ over $\mathcal{V}$. The per-token cross-entropy loss is

$$\mathcal{L}_t(x_t) = -\sum_{x \in \mathcal{V}} \delta(x - x_t) \ \log P(x \mid \mathbf{x}_{<t}; \mathcal{M}) = -\log P(x_t \mid \mathbf{x}_{<t}; \mathcal{M}),$$

where $\delta(x - x_t)$ is the Dirac delta function. Collecting the T losses produces the loss sequence $\mathrm{Seq}(\mathbf{X}) = \left\{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_T\right\}$. Furthermore, $\mathcal{L}(\mathbf{x}_i; \mathcal{M}) = -\frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_t(x_t)$ is the mean loss $\forall t \in T$ (over the whole text).

2.2.2.1 Context-aware membership signals

CAMIA (11) converts every query to $\mathcal{M}$ into the loss sequence $\mathrm{Seq}(\mathbf{X})$ above and then distils multiple scalar statistics from it. These statistics - cut-off loss, token-diversity calibrated loss, slope of the loss trajectory, count-below, count-below mean, count-below previous mean, approximate entropy and Lempel-Ziv complexity, all with repetition gaps using one and two extra copies respectively, are defined as in the original paper below, with accompanying descriptions.

1. **Cut-off loss $f_{\mathbf{Cut}}$**

$$f_{\mathrm{Cut}}(\mathbf{X}) = \frac{1}{T'} \sum_{t=1}^{T'} \mathcal{L}_t(x_t).$$

The cut-off loss signal represent a filtering out of a non-informative portion of the next-token loss sequence. The rationale is that the differences in losses are claimed to be generally larger between members and non-members early in the sequence, whereas there are large overlaps for the last indices.

2. **Token-diversity calibrated loss $f_{\mathbf{Cal}}$**

$$d_{\mathbf{X}} = \frac{|\operatorname{Dedup}(\mathbf{X})|}{|\mathbf{X}|}, \qquad f_{\mathrm{Cal}}(\mathbf{X}) \;=\; \frac{\mathcal{L}(\mathbf{X};\mathcal{M})}{d_{\mathbf{X}}}.$$

Dedup($\mathbf{X}$) is an operation which removes duplicated tokens of an input text $\mathbf{X}$. The logic behind the signal is that a text that repeats tokens is inherently easier to memorize. Dividing by diversity compensates for such intrinsic simplicity, preventing the attack from mistaking low-diversity non-members for memorized samples.

3. **Slope of the loss trajectory $f_{\mathbf{Slope}}$**

$$f_{\mathrm{Slope}}(\mathbf{X}) = \frac{T'\sum_{t=1}^{T'} t\,\mathcal{L}_t(x_t) \;-\; \sum_{t=1}^{T'} t \sum_{t=1}^{T'} \mathcal{L}_t(x_t)}{T'\sum_{t=1}^{T'} t^2 \;-\; \left(\sum_{t=1}^{T'} t\right)^2}.$$

The authors claim that the slope of the line fitted with ordinary least squares (OLS) of the next-token loss sequence is steeper for members than for non-members, under the assumption that loss values drop once a model starts recognizing a memorized sample. For non-members, memorization has not occurred and the slope is therefore expected to have a flatter slope.

4. **Count-Below (fixed) $f_{\mathbf{CB}}$**

$$f_{\mathrm{CB}}(\mathbf{X}) = \frac{1}{T'}\sum_{t=1}^{T'} \mathbb{1}\Big[\mathcal{L}_t(x_t) \le \tau\Big], \qquad \tau \in \{1,2,3\}.$$

This signal represents a low-loss vote. The absolute number of very confident tokens is often larger for memorized text even when a few high-loss outliers exist.

5. **Count-Below-Mean $f_{\mathbf{CBM}}$**

$$\overline{\mathcal{L}}_{\mathbf{X}} = \frac{1}{T'}\sum_{s=1}^{T'} \mathcal{L}_s(x_s), \quad f_{\mathrm{CBM}}(\mathbf{X}) = \frac{1}{T'}\sum_{t=1}^{T'} \mathbb{1}\Big[\mathcal{L}_t(x_t) \le \overline{\mathcal{L}}_{\mathbf{X}}\Big].$$

Similar to f_{CB}, but adapts the threshold to the example at hand, counting the number of points with lower loss than the average of the particular sequence. This is less sensitive to overall text difficulty and extreme outliers.

6. **Count-Below-Previous-Mean $f_{\mathbf{CBPM}}$**

$$\overline{\mathcal{L}}_{\mathbf{X}_{<t}} = \frac{1}{t-1}\sum_{s=1}^{t-1} \mathcal{L}_s(x_s), \quad f_{\mathrm{CBPM}}(\mathbf{X}) = \frac{1}{T'}\sum_{t=1}^{T'} \mathbb{1}\Big[\mathcal{L}_t(x_t) \le \overline{\mathcal{L}}_{\mathbf{X}_{<t}}\Big].$$

Similar to f_{CBM}, but where the number of points with lower loss then the average of the losses of the particular sequence up until the currently studied point are counted.

7. **Approximate entropy $f_{\mathbf{ApEn}}$**
   Let the length-m sliding window of per–token losses be

$$u_t^{(m)} = \Big(\mathcal{L}_t(x_t),\, \mathcal{L}_{t+1}(x_{t+1}), \ldots, \mathcal{L}_{t+m-1}(x_{t+m-1})\Big)$$

   for $t = 1, \ldots, T' - m + 1$. The authors define the distance between two vectors u_t^m and $u_{t'}^m$ as the maximum difference in their corresponding components, according to the following formula:

$$d\left(u_t^m, u_{t'}^m\right) = \max_{k=1,\ldots,m} \left|\mathcal{L}_{t+k-1}\left(x_{t+k-1}\right) - \mathcal{L}_{t'+k-1}\left(x_{t'+k-1}\right)\right|.$$

   Then, the proportion of vectors within distance r is calculated for each $u_t^{(m)}$:

$$C_t^{(m)}(r) = \frac{1}{T' - m + 1} \sum_{t'=1}^{T'-m+1} \mathbb{1}\Big[d\Big(u_t^{(m)}, u_{t'}^{(m)}\Big) \le r\Big].$$

   Averaging the log-frequencies yields

$$\Phi^m(r) = \frac{1}{T' - m + 1} \sum_{t=1}^{T'-m+1} \ln C_t^{(m)}(r)$$

   and:

$$f_{\mathrm{ApEn}}(\mathbf{X}) = \Phi^m(r) - \Phi^{m+1}(r).$$

   f_{ApEn} is used to measure the frequency of repeating patterns in a next-token loss sequence, to quantify its amount of regularity and unpredictability of fluctuations. In this implementation, m=8 and r=0.8.

8. **Lempel–Ziv complexity $f_{\mathbf{LZ}}$**

$$f_{\mathrm{LZ}}(\mathbf{X}) = \mathrm{LZW}\Big(B_1, \ldots, B_{T'}\Big)$$

   where $LZW(\cdot)$ is the Lempel-Ziv compression algorithm, which measures the complexity of a sequence of finite length, through parsing it into as few distinct phrases previously not seen in the sequence as possible. B_t is the index for the bin corresponding to $\mathcal{L}_t(x_t)$.

CAMIA does not rely on a single hyper-parameter setting for each of the eight signals described above; instead the authors build multiple variations of each signal based on the following:

1. the cut-off length T' that decides how many early tokens of the loss trajectory are considered,
2. an optional conversion from cross-entropy loss to aggregate perplexity $\mathrm{PPL}(X) = \exp L(X)$,

3. $f^k_{\text{Rep}}(\mathbf{X})$ $(k \in \{1, 2\})$ which measures how the base statistic f changes after concatenating one or two additional copies of the same text.

Only a subset of families is combined with all three variations. The complete signal suite, reproduced in Table 2.1, yields 72 one-dimensional signals; empirical effectiveness of every entry can be found in Tables 8 and 9 in Appendix B of the original paper.

Table 2.1: Summary of feature families and their variations

Family	**Variations**	T'**, τ or bins**	**# features**
f_{Cut}	f_{Cut}, $f^1_{\text{Rep,Cut}}$, $f^2_{\text{Rep,Cut}}$	$T' \in \{T, 200, 300\}$	9
f_{Cal}	f_{Cal}, $f^1_{\text{Rep,Cal}}$, $f^2_{\text{Rep,Cal}}$	$T' \in \{T, 200, 300\}$	9
f_{PPL}	f_{PPL}, $f^1_{\text{Rep,PPL}}$, $f^2_{\text{Rep,PPL}}$	$T' \in \{T, 200, 300\}$	9
$f_{\text{Cal,PPL}}$	$f_{\text{Cal,PPL}}$, $f^1_{\text{Rep,Cal,PPL}}$, $f^2_{\text{Rep,Cal,PPL}}$	$T' \in \{T, 200, 300\}$	9
f_{CB}	f_{CB}, $f^1_{\text{Rep,CB}}$, $f^2_{\text{Rep,CB}}$	$T' = 200$, $\tau \in \{1, 2, 3\}$	9
f_{CBM}	f_{CBM}, $f^1_{\text{Rep,CBM}}$, $f^2_{\text{Rep,CBM}}$	$T' \in \{T, 200, 300\}$	9
f_{LZ}	f_{LZ}, $f^1_{\text{Rep,LZ}}$, $f^2_{\text{Rep,LZ}}$	bins $\in \{3, 4, 5\}$	9
f_{CBPM}	f_{CBPM}	$T' \in \{T, 200, 300\}$	3
f_{Slope}	f_{Slope}	$T' \in \{600, 800, 1000\}$	3
f_{ApEn}	f_{ApEn}	$T' \in \{600, 800, 1000\}$	3
Total			**72**

CAMIA was evaluated on the Pythia, Pythia-deduped and GPT-Neo model suites with sizes ranging from 70M to 12B on subsets from The Pile, against the LOSS, Zlib, Min-K%, Min-K%++ and Reference membership inference attacks and showcased superior performance in both settings (11).

CAMIA also composes the membership signals $\mathcal{F} = \{f_1, f_2, \dots\}$ described above, extracted from the target model $\mathcal{M}$ on an input text $\mathbf{X}$ and chooses between two tests depending on the adversary's data access: (i) a hypothesis-testing composition that uses only a small non-member set; and (ii) a learned composition trained on a small labelled member/non-member set.

2.2.2.2 CAMIA (Edgington)

In the non-member-only setting, for any single signal $f \in \mathcal{F}$, CAMIA casts membership inference as hypothesis testing. The null hypothesis is that $\mathbf{X}$ comes from the underlying data distribution (non-member), and the alternative is that $\mathbf{X}$ is a training sample (member). Using a non-member dataset $D_{\text{non-mem}}$, the attacker computes $f(\mathbf{X}_{\text{non-mem}})$ for each $\mathbf{X}_{\text{non-mem}} \in D_{\text{non-mem}}$. Without loss of generality, CAMIA assumes that when $\mathbf{X}$ is a member, $f(\mathbf{X})$ tends to be smaller than most $f(\mathbf{X}_{\text{non-mem}})$ (e.g., lower loss for members). Otherwise, the sign of the signal is flipped.

Following prior work as stated in CAMIA, the observations $\{f(\mathbf{X}_{\text{non-mem}})\}$ are treated as samples from the distribution D_f, of the signal f over the underlying data distribution. The membership evidence for a target $\mathbf{X}$ is quantified by the p–value

$$p_f(\mathbf{X}) \;=\; \mathrm{CDF}_{D_f}\Big(f(\mathbf{X})\Big) \in [0,1],$$

i.e., the (empirical) cumulative distribution function of D_f evaluated at $f(\mathbf{X})$. Intuitively, $p_f(\mathbf{X})$ is the fraction of non-member scores not exceeding $f(\mathbf{X})$. Small $p_f(\mathbf{X})$ indicates that $f(\mathbf{X})$ lies deep in the member–like tail. For example, if $f(\mathbf{X})$ is below 80% of the non-member scores, then $p_f(\mathbf{X}) = 0.2$ and $\mathbf{X}$ is considered more likely to be a member.

After obtaining $\{p_f(\mathbf{X})\}_{f\in\mathcal{F}}$, CAMIA combines them into a single scalar and thresholds the result to decide membership. Using the paper's notation, the four combination rules considered are:

$$\begin{aligned}
\text{Edgington:} \quad & f_{\text{Edgington}}(\mathbf{X};\mathcal{M}) \;=\; \sum_{f\in\mathcal{F}} p_f. \\
\text{Fisher:} \quad & f_{\text{Fisher}}(\mathbf{X};\mathcal{M}) \;=\; \sum_{f\in\mathcal{F}} \log p_f. \\
\text{Pearson:} \quad & f_{\text{Pearson}}(\mathbf{X};\mathcal{M}) \;=\; -\sum_{f\in\mathcal{F}} \log\Big(1-p_f\Big). \\
\text{Mudholkar–George:} \quad & f_{\text{George}}(\mathbf{X};\mathcal{M}) \;=\; \sum_{f\in\mathcal{F}} \log\frac{p_f}{1-p_f}.
\end{aligned}$$

Most results in the CAMIA paper are based on the Edgington tests only, although the authors acknowledge that each method can be optimal depending on setting.

For calibration, the attacker samples a small subset of non-members from the evaluation pool to estimate each CDF_{D_f}, denoted $\alpha\%$ in the paper. $\alpha = 30$ in the main experiments, but results are also shown to be relatively consistent for different values of α. The remaining data constitute the target set on which combined scores are computed and metrics (AUC and TPR@1%FPR) are reported. Importantly, only non-members are used to build CDF_{D_f}. Members are never used for calibration in this setting.

2.2.2.3 CAMIA (Logistic Regression + Group PCA)

In the second setting, the attacker has access to a small labelled set containing both members and non-members, called the "attack dataset" D_{attack}. Rather than fixing a combining rule, CAMIA learns how to combine signals.

Given the signal set $\mathcal{F}$ and target $\mathbf{X}$, CAMIA forms the feature vector

$$\mathbf{X}_{\mathcal{F}} \;=\; \Big(f_1(\mathbf{X}), f_2(\mathbf{X}), \dots\Big) \in \mathbb{R}^{|\mathcal{F}|},$$

and trains a logistic regression model to predict membership:

$$\Pr(Y = 1 \mid \mathbf{X}_{\mathcal{F}}) \;=\; \sigma(\langle \mathbf{X}_{\mathcal{F}}, w\rangle),$$

where $\sigma(z) = 1/(1 + e^{-z})$, $Y \in \{0, 1\}$ denotes membership, and w is learned by minimizing the average cross-entropy loss over $(x, y) \in D_{\text{attack}}$. At test time, the learned model is applied to X_F of each target input and the resulting scores are thresholded to produce predictions and metrics.

Many signals used in CAMIA are correlated variations of each other, categorized in the families listed in Table 2.1. To reduce redundancy while preserving predictive structure, CAMIA applies Principal Component Analysis within each signal group and uses a small number of principal components per group as inputs to logistic regression. This approach is referred to as "Group PCA". This compresses co-varying features inside a group and improves robustness. It was shown empirically in the paper that Group PCA with two components per group yielded the best improvements (11).

For this second setting, the attacker samples a small labelled subset containing both members and non-members (again α% of the available train/test pools in the paper) to form D_{attack} for fitting the logistic model (and for fitting Group-PCA projections). The remainder is used as the target dataset for evaluation, identical to the Edgington setting to ensure a fair comparison.

2.2.2.4 Logistic Regression

Logistic regression is a commonly used probabilistic model for binary classification using a linear decision boundary. (25) Given a linear regression model $z = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_p x_p = \theta^\intercal \mathbf{x}$, the logistic regression model can be retrieved by mapping z to the interval $[0, 1]$ through the logistic function $h(z) = \frac{e^z}{1+e^z}$, giving:

$$g(\mathbf{x}) = \frac{e^{\theta^\intercal \mathbf{x}}}{1 + e^{\theta^\intercal \mathbf{x}}}.$$

This can be interpreted as the probability for one of the two classes, implicitly providing the probability for the other class through $1 - g(\mathbf{x})$. The parameter θ is learned from the training data through a maximum likelihood approach, where $\mathbf{X}$ is the input and $\mathbf{y}$ is the output, by solving

$$\hat{\theta} = \arg\max_\theta \, p(\mathbf{y}|\mathbf{X}; \theta) = \arg\max_\theta \, \sum_{i=1}^{n} \ln \, p(y_i|\mathbf{x_i}; \theta).$$

2.2.2.5 Principal Component Analysis

Principal component analysis (PCA) is a common technique employed to learn a low-dimensional representation of certain data. This is achieved by the projection of the original data onto a linear subspace of a lower dimension through linear transformation, while retaining as much information as possible (in terms of variance). (25) The data $\mathbf{X}$ is first centered by subtracting the mean value from each data point, giving a matrix $\mathbf{X}_0$, to which a matrix factorization technique called singular value

decomposition is applied. The result can be used to extract the first q vectors called principal axes, which represent the optimal low-dimensional representation of $\mathbf{X}_0$.

Note that it is not described in (11) how the implementation of logistic regression or PCA is done in detail, leaving technical details such as the optimization method unknown.

2.2.3 Membership Inference Attacks via Neighbourhood Comparison

2.2.3.1 Problem setting

Let $f_{\boldsymbol{\theta}}$ be a language model trained on an unknown dataset D_{train}. The adversary can query $f_{\boldsymbol{\theta}}$ in a grey-box fashion, obtaining the (token-level) negative log-likelihood $\mathcal{L}(f_{\boldsymbol{\theta}}, x)$ of any text $x \in \mathcal{X}$, but has no access to model weights or gradients. The goal is to decide, for a given x, whether $x \in D_{\text{train}}$. Instead of contrasting x with an external reference model, the Membership Inference Attack via Neighbourhood Comparison (hereinafter referred to as the Neighbourhood Comparison Attack or NCA) (16) compares x to a set of n synthetic neighbours $\mathcal{N}(x) = \{\tilde{x}_1, \ldots, \tilde{x}_n\}$ that have been crafted to be semantically and syntactically almost identical to x while not belonging to D_{train}. Membership is inferred with the indicator

$$A_{f_{\boldsymbol{\theta}}}(x) \;=\; \mathbb{1}\left[\Big(\mathcal{L}(f_{\boldsymbol{\theta}}, x) \;-\; \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(f_{\boldsymbol{\theta}}, \tilde{x}_i)\Big) \;<\; \gamma\right],$$

where the threshold γ is calibrated to meet a predetermined low false positive rate. Intuitively, for non-members the loss of x should be comparable to the average loss of its near-equivalent neighbours, whereas genuine training examples tend to receive an unusually low loss, making the difference (in brackets) negative and if it crosses γ triggering a positive membership claim.

2.2.3.2 Neighbour generation

NC constructs $\mathcal{N}(x)$ entirely without access to D_{train}. Following the lexical-substitution strategy of (30), it uses an off-the-shelf masked-language model (MLM) such as BERT to propose context-appropriate word replacements. Algorithm 1 reproduces the exact procedure, where the probability $p_\theta(\tilde{w} = w^{(i)}|x)$ of token $\tilde{w}$ as the word in position i can be obtained from the masked language model's probability distribution $p(\mathcal{V}^{(i)}|x)$ over the token vocabulary $\mathcal{V}$ at position i. (Algorithm 1 in (16)).

Algorithm 1 Neighbourhood Generation (from 16)

Require: Text $x = (w^{(1)}, \ldots, w^{(L)})$, integers n (neighbours) and m (replacements)
Ensure: Neighbours $\{\tilde{x}_1, \ldots, \tilde{x}_n\}$, each with m word replacements

1: **for** $i \in \{1, \ldots, L\}$ **do**
2: Get token embeddings $\Big(\phi(w^{(1)}), \ldots, \phi(w^{(L)})\Big)$
3: Add dropout to position i: $\phi(w^{(i)}) \leftarrow \text{drop}\Big(\phi(w^{(i)})\Big)$
4: Obtain $p\Big(\mathcal{V}^{(i)} \mid x\Big)$ from BERT
5: Compute $p_{\text{swap}}\Big(w^{(i)}, \tilde{w}^{(i)}\Big) \forall \tilde{w} \in \mathcal{V}$ via

$$p_{\text{swap}}\Big(\hat{w}^{(i)}, \tilde{w}^{(i)}\Big) = \frac{p_\theta\Big(\tilde{w} = w^{(i)} \mid x\Big)}{1 - p_\theta\Big(\hat{w} = w^{(i)} \mid x\Big)}$$

▷ Eq. (4) in 16

6: For all swaps $(w^{i_1}, \tilde{w}^{i_1}), ..., (w^{i_m}, \tilde{w}^{i_m})$ with $i_k \neq i_l$ for $i \neq l$, compute joint suitability $\sum_{i=1}^{m} p_{\text{swap}}\Big(w^{(i_1)}, \tilde{w}^{(i_1)}\Big)$
7: **return** the n candidates with highest joint suitability

Because each $\tilde{x}_i$ differs from x by only a few high-probability token swaps, all neighbours lie close to x under the language distribution. Thus, any substantial loss gap is attributable to $f_{\boldsymbol{\theta}}$ having memorised x during training, exactly the signal NCA exploits.

2.2.4 Semantic Membership Inference Attack (SMIA)

Most prior MIAs against LLMs look for exact substrings that the model may have seen during training. Such verbatim evidence is only a narrow slice of the privacy surface: large language models frequently encode the meaning of a passage without ever reproducing it word-for-word. The core idea behind SMIA (13) is therefore to probe an LLM not with a single text x but with a constellation of semantically nearby alternatives. If x was a training sample, the model will already have aligned its internal representations with the underlying idea; it should react to small semantic shifts in a smoother and more predictable way than it does for a genuinely novel concept.

Formally, let T be the black-box target model whose training data is to be tested, and let $\ell(., T)$ denote its loss value or log probability on any input text. SMIA assumes access to two auxiliary corpora:

$$D_{\text{tr–m}} \subseteq \text{training data of } T, \qquad D_{\text{tr–n}} \cap D_{\text{tr–m}} = \varnothing,$$

where the second set is drawn from the same domain but guaranteed to be unseen by T (e.g. Wikipedia articles published after the model's cut-off date). From these corpora the attack learns to tell members from non-members in four steps.

1. **Semantic perturbations:** For every $x \in D_{\text{tr–m}} \cup D_{\text{tr–n}}$ the authors created n neighbours by masking K random tokens and letting a text-infilling model $N(\cdot)$ (T5 with 3B parameters in the paper) replace them through substitution, deletion or duplication.

2. **Measuring semantic distance:** The Cohere v3 encoder E maps any text to a 1024-dimensional embedding $\phi(x)$. The differences in semantic vectors

$$\Delta\phi(x, \tilde{x}) \;=\; \phi(x) - \phi(\tilde{x}).$$

   can then be calculated.

3. **Capturing the model's behavioural change:** Alongside the embedding shift the loss difference

$$\Delta\ell(x, \tilde{x}) \;=\; \ell(x, T) - \ell(\tilde{x}, T).$$

   can also be calculated.

4. **Learning a decision boundary:** Each pair $\big(\Delta\phi,\, \Delta\ell\big) = \big(\phi(x) - \phi(\tilde{x}),\, \ell(x, T) - \ell(\tilde{x}, T)\big)$ constitutes one training vector in $\mathbb{R}^{1025}$. SMIA collects an equal number of such vectors from $D_{\text{tr–m}}$ (label 1) and $D_{\text{tr–n}}$ (label 0) and trains a small neural network model $A(\cdot)$ for 20 epochs with the Adam optimizer set at $r = 5 \times 10^{-6}$) or $r = 1 \times 10^{-6}$, depending on setting.

Given an unseen candidate x,n_{inf} new neighbours are generated, and the corresponding feature vectors are computed and passed through the trained neural network:

$$\mu \;\leftarrow\; \frac{1}{n} A\Big(\Delta\phi(x, \tilde{x}^{(i)}), \Delta\ell(x, \tilde{x}^{(i)})\Big), \qquad i = 1, \ldots, n_{\text{inf}}.$$

The final membership score is determined based on whether $\mu > \varepsilon$, where ε is a predetermined threshold and a μ being greater infers membership and vice versa.

SMIA outperforms all membership inference attacks it was compared to (LOSS, Ref, Zlib, Nei, Min-K and Min-k++) in terms of both the AUC-ROC and TPR@1%FPR metrics. The authors claim that these results stem from the incorporation of semantics into the analysis as opposed to other method's reliance on the target model's behaviour alone, and the fact that a neural network is trained particularly to distinguish between members and non-members.

2.2.5 Range Membership Inference Attacks (RaMIA)

RaMIA (12), extends membership inference to a range setting, where a challenger challenges an adversary to discern whether any point within a specified range lies in the training set. This means that multiple data points can be searched for, instead of a single one. Below is the formulation of the Range Membership Inference Game from the paper:

Range Membership Inference Game (Definition 2 in (12)).

Let π be the data distribution, and let T be the training algorithm.

1. *The challenger samples a training dataset $D \leftarrow^{s_D} \pi$, and trains a machine learning model $\theta \leftarrow T(D)$.*
2. *The challenger samples a data record $z_0 \leftarrow^{s_{z0}} \pi$ from the data distribution, and a training data record $z_1 \leftarrow^{s_{z1}} D$.*
3. *The challenger flips a fair coin to get the bit $b \in \{0, 1\}$. If $b = 1$, the challenger samples a range $\mathcal{R}_1$ containing at least one training point. Otherwise, challenger samples a range $\mathcal{R}_0$ containing no training points.*
4. *The challenger sends the target model θ and the range $\mathcal{R}_b$ to the adversary.*
5. *The adversary gets access to the data distribution π and access to the target model, and outputs a bit $\hat{b} \leftarrow A(\theta, \mathcal{R}_b)$.*
6. *If $\hat{b} = b$, output 1 (success). Otherwise, output 0.*

When moving from a single data point x to a range $\mathcal{R}$, the standard membership hypotheses become:

$$H_0 : \forall z \in \mathcal{R},\, z \notin D \quad \text{vs.} \quad H_1 : \exists z \in \mathcal{R} \text{ s.t. } z \in D.$$

Since $\mathcal{R}$ can contain infinitely many points, the authors opt for sampling a set $S \subseteq \mathcal{R}$. Then the hypotheses become

$$H_0 : \forall z \in S,\, z \notin D \quad \text{vs.} \quad H_1 : \exists z \in S \text{ s.t. } z \in D.$$

To test θ under H_0 or H_1, (12) discuss Bayes Factor and the Generalized Likelihood Ratio Test (GLRT) as standard statistical approaches for these "composite" hypotheses. However, they also highlight practical complications due to imperfect MIA scores and out-of-distribution samples.

As a simple strategy to handle spurious outliers or unreliable membership scores, (12) propose the following "trimmed" approach (their Eq. (4)):

$$P(\theta \mid H_1) \;=\; \text{TrimmedAvg}\Big(S, q_s, q_e; P\Big) \;=$$

$$\text{Avg}\Big\{\, P(\theta \mid x \in D) \;\Big|\; x \notin [q_s, q_e]\text{-quantiles of } P(\theta \mid x \in D)\Big\},$$

where S is the sampled set from $\mathcal{R}$, and q_s, q_e define quantiles to remove from the top or bottom. For instance, if the adversary suspects that top-scoring points might be out-of-distribution data with artificially high membership scores, they can trim those to produce a more robust membership statistic.

RaMIA is a framework wherein:

1. A sampler, Sample($\mathcal{R}$), draws candidate points from the range $\mathcal{R}$.
2. Any existing point-based membership tester with membership scoring function MIA(x) is applied to each sampled point x.

By combining these components, an adversary derives a range membership score and predicts whether $\mathcal{R}$ overlaps with the training data. Concretely, one may compute $\text{TrimmedAvg}\big(S, q_s, q_e; \text{MIA}\big)$ (or a Bayes Factor / GLRT) and compare against a threshold. Thus, RaMIA offers a general mechanism to capture privacy risks that arise whenever any point in a region of the input space belongs to D. This can reveal memorization that goes beyond the single "point query" scope of conventional MIAs.

3

Methods

3.1 Workflow overview

The work was carried out through an iterative process consisting of literature review, design, and implementation. The first step was reviewing recent research on membership inference attacks, focusing on papers such as CAMIA, SMIA, RaMIA, and the Neighbourhood Comparison Attack. Weekly discussions with the advisors at AI Sweden and the supervisor helped guide the selection of approaches.

CAMIA was chosen as the main starting point because of its strong reported performance and modular architecture, which makes it suitable for adding new signals. SMIA was considered but excluded later due to its computational cost and its reliance on both member and non-member samples. RaMIA was also deferred since it requires an existing attack as input. The Neighbourhood Comparison Attack was selected as an additional attack because it is computationally lighter and arguably preserves semantics better than SMIA's neighbour generation, and can easily be treated as a signal in the CAMIA architecture.

The implementation, experiments, and design choices were also developed iteratively throughout the project, with frequent adjustments based on results and practical constraints such as compute resources.

3.2 Custom CAMIA implementation

3.2.1 Implementation methodology

The paper Context-Aware Membership Inference Attacks against Pre-trained Large Language Models (11) was published on 11 September 2024 and was under review during the project. The authors received a request for their code, but they declined due to the ongoing review. They instead pointed to the public MIMIR benchmark codebase and noted that CAMIA is built on that architecture. By the time they replied, a from-scratch re-implementation of CAMIA had already started. Adopting MIMIR at that point would have added dependencies and complexity, and it was unclear how to integrate the planned extensions. The implementation therefore proceeded independently, based on the description in the original paper (11). Building the attack from scratch was time-consuming and required several rounds of debugging

and sanity checks of the outputs.

A second bottleneck was the availability of data. The tests run in the CAMIA paper were based on datasets from The Pile (19), a large-scale open source repository of diverse text modality data sources. However, the original datasets had been taken down due to copyright strikes (18), and could not be retrieved from GitHub, Hugging Face or similar easily. However, it was later found that the tests run on the MIMIR benchmark also used data from The Pile, and portions of the subsets used for CAMIA tests can be streamed directly from the MIMIR repository (15). These subsets have not been subject to copyright claims, and can thus be used without any known legal considerations. As the language models used for the experiments are also open source and publicly available, with the infrastructure provided by AI Sweden, all conditions for a faithful custom implementation were fulfilled.

Not all technical details necessary for a one-to-one implementation are clear or described in (11), so a number of independent assumptions and design choices were made. Notwithstanding these potential differences, the results reported in the paper were essentially reproduced for the models and datasets experimented on. Full results are reported in the Results chapter.

3.3 Neighbourhood Comparison Attack signal

A central purpose of the thesis is to further develop recent MIAs to improve research in the field. Once the notebook with the custom CAMIA implementation generated results similar to the original paper (see Chapter 4 for full results), the next part of the design process consisted of choosing and implementing an additional signal utilizing semantic information of the available text samples. The methodology described in Membership Inference Attacks against Language Models via Neighbourhood Comparison (16) was chosen over SMIA due to its lightweight and reference-free approach suitable both to the Edgington (non-member only) and logistic regression with principal component analysis approach (members and non-members) in CAMIA. An additional benefit was that the code for the Neighbourhood Comparison Attack was publicly available (17), which decreased the coding workload. In order to make the neighbour signal work, two major steps were required:

1. Generating neighbours for the arXiv, dm_mathematics, github, pubmed_central, hackernews and pile_cc data subsets (see Section 3.5 for details); and
2. Converting the Neighbourhood Comparison Attack into a signal which could be integrated into the CAMIA architecture.

3.3.1 Neighbour Generation

Both steps were relatively straightforward, as the code for the Neighbourhood Comparison Attack included code for generating neighbours. Relevant code fragments were incorporated into cells in the custom CAMIA implementation notebook, and neighbours were generated according to Algorithm 1. As generating neighbours is too time-consuming to execute every time an experiment is run, they were created

once and cached for subsequent experiments. 100 neighbours per member and non-member were generated for each dataset, as the experiments in the original paper were performed on up to 100 neighbours. In other words, n was set to 100. m, the number of replacements required as an argument by the algorithm represent the number of tokens to be exchanged per neighbour. Since the authors had already conducted extensive experiments on the efficiency of various values for m and found that a single word replacement was superior to two or more word replacements in all studied cases (see Figure 3.1), m was set to 1 in the implementation to reduce the number of hyper-parameters and sets of neighbours to a practically feasible small set of experiments.

#Word Replacements	1	2	3
News:			
1% FPR	8.29%	4.09%	4.18%
0.1% FPR	1.73%	0.85%	0.94%
0.01% FPR	0.29%	0.23%	0.21%
Twitter:			
1% FPR	7.35%	4.86%	4.37%
0.1% FPR	1.43%	0.74%	0.72%
0.01% FPR	0.28%	0.14%	0.11%
Wikipedia:			
1% FPR	2.32%	1.76%	1.44%
0.1% FPR	0.27%	0.23%	0.17%
0.01% FPR	0.10%	0.07%	0.03%

Figure 3.1: Attack performance w.r.t. the number of words (Table 5 in (16))

It should here be noted that the p_{swap}-based selection in the Neighbour Generation algorithm is in turn adopted from Zhou et al. (2019), and represents a way of maintaining both semantic equivalence and preserving syntax after token perturbation. A transformer-based masked language model such as BERT is used to obtain the probability distribution for the token vocabulary $\mathcal{V}$ at position i. Instead of merely masking the token to be exchanged, which would obfuscate the meaning of the sentence and thus losing semantics, strong dropout is added to the input embedding layer at position i before it is fed into the transformer to achieve the candidate replacements for $w^{(i)}$. This method arguably preserves semantics to a greater extent than the neighbour generation procedure in SMIA, which both deletes and duplicates random tokens in some cases.

3.3.2 Delta-loss Statistic

Given the simplicity and reference-free nature of the Neighbourhood Comparison Attack, where the difference between the loss of a target sample and the average loss of its neighbours is compared against a threshold, it is easy to incorporate as a signal in the CAMIA architecture where the other signals are already based on the next-token loss sequence from a target large language model for a specific sample, in both the Edgington (non-members only) and LR + PCA (members and non-members) cases.

A novel contribution which was added to the architecture here is based on the fact that the neighbourhood delta-loss statistic is a single signal, which can easily drown out

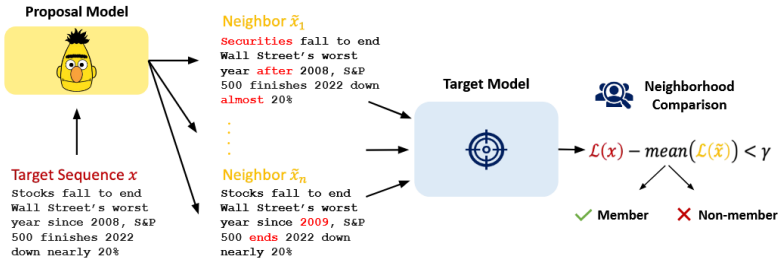


Figure 3.2: Overview of the Neighbourhood Comparison Attack (Figure 1 in (16))

among the 72 other CAMIA signals. In order to acknowledge that the neighbour signal essentially is a stand-alone attack, it was magnified to increase its influence compared to other signals. Instead of duplicating the neighbour signal through repetitions and truncations similar to the original CAMIA signals, the signal was amplified through a scalar weight. Hyperparameter grid searches were conducted with the p-value in the Edgington (and other p-value statistics tests) having a weight from the set $\{0, 9, 18, 36, 72\}$, in the Edgington test, and a feature weight from the set $\{0, 1, 2, 3, 4\}$ in the logistic regression test. As the two tests are independent, the hyperparameter search used one weight from each set simultaneously (e.g. 9 for Edgington and 1 for logistic regression). Both weights were also set to 0 in some experiments to also assess performance without incorporating the neighbour signal. Another hyperparameter is the amount of neighbours n used, where $n \in \{0, 25, 100\}$. In the original Membership Inference Attacks against Language Models via Neighbourhood Comparison paper, experiments were conducted with $n \in \{5, 10, 25, 50, 100\}$. As the results in the authors' experiments showed that performance only increased or at least remained as strong as the amount of neighbours increased (see Figure 3.3), the number of tests were reduced in this implementation as the impact of the other hyperparameters was more uncertain.

#Neighbours	5	10	25	50	100
News:					
1% FPR	2.98%	4.57%	6.65%	8.19%	8.29%
0.1% FPR	0.53%	0.79%	1.43%	1.50%	1.73%
0.01% FPR	0.05%	0.07%	0.18%	0.23%	0.29%
Twitter:					
1% FPR	3.93%	4.88%	6.21%	6.63%	7.35%
0.1% FPR	0.57%	0.62%	1.01%	1.34%	1.43%
0.01% FPR	0.05%	0.07%	0.10%	0.23%	0.28%
Wikipedia:					
1% FPR	1.57%	1.81%	2.02%	2.17%	2.32%
0.1% FPR	0.16%	0.21%	0.23%	0.26%	0.27%
0.01% FPR	0.05%	0.08%	0.09%	0.10%	0.10%

Figure 3.3: Attack performance w.r.t. the number of neighbours (Table 4 in (16))

3.4 Loss-Guided Semantic Neighbours

Combining CAMIA with the Neighbourhood Comparison Attack provides novel opportunities for generating semantic neighbours, based on the next-token loss sequences. Since a fundamental rationale of CAMIA is to utilize more available information from target models and samples than historical membership inference attacks, it makes sense to guide the perturbation of semantic neighbours utilizing the same new set of information. The intuition that high loss tokens carry more relevant information for inference, can be formalized in the following hypothesis: MIA performance will improve with the generation of semantic neighbours, based on perturbations of text samples at token indices with the greatest loss in the sequence. A second set of neighbours was generated based on this approach, where for the K highest loss tokens, a single neighbour was generated for each corresponding index with the otherwise identical Algorithm 1 used for the original set. Specifically, the adjusted algorithm based on the K highest-loss-indices heuristic can be represented in the following manner:

Algorithm 2 Neighbourhood Generation (from 16)

Require: Text $x = (w^{(1)}, \ldots, w^{(L)})$, integers n (neighbours), m (replacements), and K (number of highest loss indices), where $L \geq K$

Ensure: One neighbour $\{\tilde{x}_k\}$ with m word replacements per position $k \in S_K$, where S_K is the set of indices corresponding to the K highest losses in the next-token loss sequence of x

1: $S_K \leftarrow$ indices of the K largest losses in $\{\mathcal{L}_i\}_{i=1}^{L}$
2: **for each** $k \in S_K$ **do**
3: Get embeddings $\Big(\phi(w^{(1)}), \ldots, \phi(w^{(L)})\Big)$
4: Add dropout: $\phi(w^{(k)}) \leftarrow \text{drop}\Big(\phi(w^{(k)})\Big)$
5: Obtain $p\Big(\mathcal{V}^{(k)} \mid x\Big)$ from BERT
6: Compute $p_{\text{swap}}\Big(w^{(k)}, \tilde{w}^{(k)}\Big) \forall \tilde{w} \in \mathcal{V}$ via

$$p_{\text{swap}}\Big(\hat{w}^{(k)}, \tilde{w}^{(k)}\Big) = \frac{p_\theta\Big(\tilde{w} = w^{(k)} \mid x\Big)}{1 - p_\theta\Big(\hat{w} = w^{(k)} \mid x\Big)}$$

▷ Eq. (4) in 16

7: $\tilde{w}^{(k)} \leftarrow \arg\max_{\tilde{w} \in \mathcal{V}} p_{\text{swap}}(w^{(k)}, \tilde{w}^{(k)})$
8: $\tilde{x}_k \leftarrow$ replace $w^{(k)}$ in x with $\tilde{w}^{(k)}$ ▷ one neighbour per high-loss index
9: **return** $\{\tilde{x}_k \mid k \in S_K\}$

3.4.1 Motivation from Next-Token Loss Trajectories

To further guide more involved generation of semantic neighbours for utilization in CAMIA, an attempt was next made to extract indices in the next-token loss sequence where perturbations potentially could have the largest differential effect on the loss of the resulting set of neighbours.

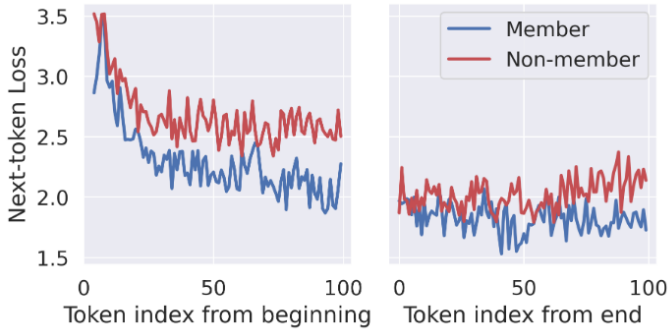


Figure 3.4: Average next-token loss as a function of token index (Figure 3 in (11))

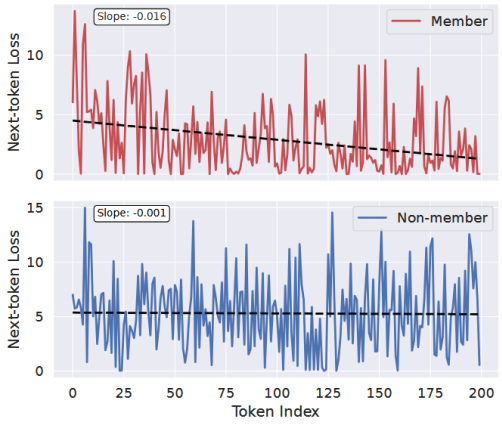


Figure 3.5: Linear functions fit to next-token loss sequences (Figure 4 in (11))

The authors of (11) argue that the average loss difference between members and non-members per token index is on average larger in the earlier parts of a sequence 3.4. Furthermore, they also claim that the average slope is steeper for members than for non-members 3.5 (which guided the formalization of f_{Slope} in the first place). An argument around these points raised in the paper is *"The intuitive explanation is that as the model continues to predict the next tokens, the member (i.e., a sample from the training data) would seem more and more familiar to the model whereas for non-members, the model's uncertainty in next-token prediction remains"*. (11)

For these reasons, the following additional hypotheses are formed:

1. Regions of drastic change at an aggregate level in next-token loss sequences carry useful information for membership inference attacks;

2. The first index in a next-token loss sequence from which the slope of the ordinary least squares (OLS) fitted line between the same index and the last is sufficiently close to zero carries useful information for membership inference attacks; and

3. A set of indices within radius ϵ of such an index is a feasible selection for generating a set of semantic neighbours to be used for an improved neighbourhood comparison signal.

The first hypothesis above leads to the following proposed novel next-token loss sequence based signal.

3.4.2 Elbow-Point Heuristic

The authors of (11) claim that the rate of change in loss for members is larger than for non-members (see Figure 3.5). They also observe that the loss difference between members and non-members is on average larger in the earlier parts of a sequence (see Figure 3.4). Together these facts indicate that the rate of change in loss for members on average is larger in the beginning of a sequence than in the end. For an index t_m, it could therefore often be expected that a prefix fitted between indices 0 and t_m has a steeper slope than a suffix fitted between t_m and T (the last index). There also in general exist sequence-specific arg max indices t_m^* and t_n^* for each member and non-member respectively, where the difference between slopes of prefixes and suffices is maximized. Given the observations mentioned above, it should be expected that the magnitudes of the prefix and suffix slope differences to be significantly different between members and non-members. In other words, since the difference of the slopes must be maximized for some index, there exist optimal "elbow points" t_m^* and t_n^* for members and non-members, with different prefix and suffix characteristics. The values of t_m^* and t_n^* should also, relative to sequence length T, on average be different.

These arguments form the basis for the formalization of two definitions of two novel signals for the attack architecture. Let

$$\mathcal{D}_{pad} \;=\; \Big\{ t_e \;\Big|\; t_e \geq \tau_{\mathrm{L}} \;\wedge\; t_e \leq T - \tau_{\mathrm{R}} \Big\}, \qquad \tau_{\mathrm{L}} = 25,\; \tau_{\mathrm{R}} = 0.25\,T\;,$$

and for any pivot index t_e ($1 \leq t_e \leq T-1$) define

$$\begin{aligned} P_{t_e}: &\quad \text{OLS line fitted to } (i, \mathcal{L}_i(x_i)) \text{ for } i = 1, \ldots, t_e, \\ S_{t_e}: &\quad \text{OLS line fitted to } (i, \mathcal{L}_i(x_i)) \text{ for } i = t_e, \ldots, T, \end{aligned}$$

where $\frac{\delta P_{t_e}}{\delta T}$ and $\frac{\delta S_{t_e}}{\delta T}$ are denoted by $a_P(t_e)$ and $a_S(t_e)$ (OLS is an abbreviation for ordinary least squares, a method described in the third point of 2.2.2.1). Then, define the following two signals:

1. **Relative optimal elbow point index,** *elbow_frac*:

$$\boxed{\texttt{elbow\_frac}(X) \;=\; e_f^* = \frac{t_e^\star}{T}} \quad \text{where} \quad t_e^\star = \underset{t_e \in \mathcal{D}_{pad}}{\arg\max} \Big| a_P(t_e) - a_S(t_e) \Big|.$$

2. **Absolute magnitude of maximal slope difference,** *elbow_delta_slope*:

$$\boxed{\texttt{elbow\_delta\_slope}(X) \;=\; e_s^* = \Big| a_P(t_e^\star) - a_S(t_e^\star) \Big|}$$

The domain D_{pad} is constructed to create padding with an absolute number of tokens in the beginning, and a relative number of tokens in the end. τ_L is set to avoid

the influence of variance by fitting the prefix to too few tokens which may generate non-representative slopes. τ_R is set to not consider the last portion of the sequence, as initial experiments showed that there in general are two main clusters of optimal elbow points - at the beginning and the end of next-token loss sequences. Such a bimodal distribution is not easily extendable to this setting, where the goal is to distinguish unimodal distributions of members and non-members. Furthermore, the intuition under the hypothesis leading to the design of these signals, is that the loss signal for members decrease once memorization starts to make an effect. Therefore the first cluster is chosen for the elbow signals. Concerning the absolute magnitude of maximal slope difference signal, the absolute value is utilized also to avoid generating a bimodal distribution, as it cannot be assumed that the prefix is always steeper than the suffix - particularly not for non-members for which the slope in (11) is claimed to generally be flat.

Explicitly, for any pivot index $t_e \in \mathcal{D}_{pad}$ the OLS slope of the prefix and suffix lines can be written with running sums:

$$a_P(t_e) \;=\; \frac{t_e \sum_{i=1}^{t_e} i\,\mathcal{L}_i(x_i) \;-\; \Big(\sum_{i=1}^{t_e} i\Big)\Big(\sum_{i=1}^{t_e} \mathcal{L}_i(x_i)\Big)}{t_e \sum_{i=1}^{t_e} i^2 \;-\; \Big(\sum_{i=1}^{t_e} i\Big)^2},$$

$$a_S(t_e) \;=\; \frac{(T-t_e+1)\sum_{i=t_e}^{T} i\,\mathcal{L}_i(x_i) \;-\; \Big(\sum_{i=t_e}^{T} i\Big)\Big(\sum_{i=t_e}^{T} \mathcal{L}_i(x_i)\Big)}{(T-t_e+1)\sum_{i=t_e}^{T} i^2 \;-\; \Big(\sum_{i=t_e}^{T} i\Big)^2}.$$

These formulae evaluate in $O(1)$ time once the three running sums $\sum i$, $\sum i^2$ and $\sum i\,\mathcal{L}_i(x_i)$ are available, so the full scan over all $t_e \in \mathcal{D}_{pad}$ remains $O(T)$.

For completeness, it should be noted that the relative optimal elbow point index is chosen as a potentially relevant signal rather than the absolute index, simply because different samples have different token lengths T, and since the signal should be generally applicable.

3.4.3 Flatline-Point Heuristic

One way to verify whether the intuitive explanation in (11) about a target model's familiarity with a member sample leads to more pronounced decreasing average loss than for a non-member, is to design a signal which measures when the derivative of the OLS line fitted to next-token loss signal approaches 0, or in other words when it starts to "flatline". Given the claim that the slope for non-members is generally flat, it should be expected for most non-members to have its OLS fitted line having a slope close to 0. This should in turn trivially mean that for non-members the first index t_f in the next-token loss sequence from which the OLS fitted line has a slope close to 0 is the first one. On the other hand, members could instead be expected to exhibit a

decreasing average loss up to a critical point where memorization starts to make a significant impact. After this point, the hypothesis is that the slope of the fitted OLS line of the remaining part of the sequence remains relatively constant, as the model is no longer as "surprised" by subsequent tokens, leading to on average constant loss. This can be conceptualized as the signal "flatlining". If these assumptions are true, it should be expected that the location from which the OLS fitted line's slope is sufficiently close to 0 being different for members and non-members. This can also be formalized as an additional novel signal:

1. **Relative optimal flatline point index,** $flat\_frac$**:** Let $\varepsilon_0 > 0$ be the initial tolerance and $\varepsilon_{\max}$ the upper bound ($\varepsilon_0 = 10^{-3},\ \varepsilon_{\max} = 8\,\varepsilon_0$ in all experiments). Define the smallest workable tolerance

$$\varepsilon^\star \;=\; \min\Big\{\, \varepsilon \in [\varepsilon_0, \varepsilon_{\max}] \;\Big|\; \exists\, t \in D_{T-1} :\ |a_S(t)| \le \varepsilon \Big\}.$$

   Using that $\varepsilon^\star$, choose the first suffix whose OLS slope is sufficiently flat:

$$t_f^\star \;=\; \min\Big\{\, t_f \in \mathcal{D}_{T-1} \;\Big|\; |a_S(t_f)| \le \varepsilon^\star \Big\}, \qquad \boxed{\texttt{flat\_frac}(X) = f^\star = \frac{t_f^\star}{T}}$$

If no t_f^* exists according to the definition above, a fallback solution is to instead choose $t_f^\star = \arg\min_{t \in \mathcal{D}_{T-1}} |a_S(t)|$. This is the index from which the suffix has the flattest slope. Note that this definition is not used as a first choice, since the interesting region is where the loss signal starts to flatline - which is not necessarily where the suffix slope is the smallest.

3.4.4 Next-token loss sequence guided semantic neighbour generation

As a final design choice in this thesis, an attempt to utilize the new signals extracted from the next-token loss sequence to guide targeted semantic neighbour generation followed. Using the intuition that the next-token loss sequence starts flatlining when memorization occurs, focusing token perturbations around this region could potentially "revive" the next-token loss sequence, under the assumption that the loss for each token is dependent on the previous part of the sequence (22). Therefore, two additional experiments where the neighbourhood comparison signal is introduced to the custom CAMIA implementation based on sets of neighbours with optimal elbow and flatline indices for a specific dataset on a target large language model are proposed. A crucial consideration is how many neighbours should be generated per token index at or in the vicinity of the optimal point of the chosen signal. Although an optimal point is searched for, exclusively perturbing a single token with the kind of semantics-preserving algorithm utilized in (16), would (under the condition that each neighbour is unique) essentially require that 100 unique synonyms or otherwise semantics-preserving tokens are available in the token vocabulary, which is hardly realistic. Furthermore, introducing slack around the optimal point could in itself be beneficial for at least two reasons:

- If the tokenizers for extracting the next-token loss signals and neighbour generation are different, there is no guarantee that the optimal index corresponds to the same location after tokenization in the second tokenizer (24); and

- If memorization starts to occur from where the slope of a next-token loss sequence approaches 0, it should intuitively not in general be restricted to a single token, but rather the general (small) region around the optimal point.

To simultaneously acknowledge the potential importance of the optimal points under the hypotheses the elbow and flatline signals are based on, a method to select the n indices is to fit a Gaussian around the optimal point with appropriate variance, to include a limited number of points in the vicinity with a decreasing number of neighbours as the distance from the optimal point increases.

The goal is to allocate an integer neighbour budget n_i to each token position i in a symmetric window $\{t^\star - M, \dots, t^\star + M\}$ around a pivot index $t^\star$ such that:

1. The discrete histogram $\{n_i\}$ follows a normal-distribution shape $g(i) = \exp\Big(-(i - t^\star)^2/2\sigma^2\Big)$;

2. The counts sum exactly to the requested total $\sum_i n_i = N_{\text{tot}}$; and

3. If an offset pushes i outside the valid token span $[1, L-2]$[1], its quota redistributed proportionally over the remaining in-range indices, preserving the relative Gaussian shape.

The algorithm should ideally do the following:

- Compute the ideal real-valued shares $\tilde{w}_d = g(t^\star + d)$ for all offsets $d \in \{-M, \dots, M\}$ and normalise them.

- Convert to provisional integer counts $\hat{n}_d = \text{round}(\tilde{w}_d N_{\text{tot}})$, correcting any ± 1 rounding drift.

- Since some d may be out of bounds, split offsets into *valid* vs. *invalid*; accumulate the lost quota $n_{\text{lost}} = \sum_{\text{invalid}} \hat{n}_d$

- Re-allocate n_{lost} to the valid offsets in proportion to their original Gaussian shares and round once more so the final integer vector $\{n_i\}$ still sums to N_{tot}

- Finally, for each in-range index i generate n_i semantic neighbours using the standard dropout + masked-LM p_{swap} routine of (11).

Formally, Algorithm 1 can then be updated to:

[1]Tokens are indexed so that position 0 is the BOS token and position $L-1$ is the final token before EOS, hence the usable range is $1 \dots L-2$.

Algorithm 3 Neighbourhood Generation with Gaussian Boundary Reallocation

Require: Text $x = (w^{(1)}, \ldots, w^{(L)})$, pivot index $t^\star$, integers N_{tot} (total neighbours) and m=1, Gaussian bandwidth σ, maximum offset M
Ensure: Neighbours $\{\tilde{x}_1, \ldots, \tilde{x}_{N_{\text{tot}}}\}$, each with one word replaced

$\mathcal{D} \leftarrow \{-M, \ldots, M\}$ ▷ candidate offsets
$g_d \leftarrow \exp\!\left(-d^2/2\sigma^2\right)$ for $d \in \mathcal{D}$
$\tilde{w}_d \leftarrow g_d / \sum_{j \in \mathcal{D}} g_j$
$\hat{n}_d \leftarrow \text{round}(\tilde{w}_d N_{\text{tot}})$
Correct rounding drift so $\sum_d \hat{n}_d = N_{\text{tot}}$
$\mathcal{D}_{\text{valid}} \leftarrow \{\, d \in \mathcal{D} \mid 1 \le t^\star + d \le L - 2\}$
$n_{\text{lost}} \leftarrow \sum_{d \notin \mathcal{D}_{\text{valid}}} \hat{n}_d$
if $n_{\text{lost}} > 0$ **then**
    Re-scale $\tilde{w}_d \leftarrow \tilde{w}_d \Big/ \sum_{j \in \mathcal{D}_{\text{valid}}} \tilde{w}_j$ for $d \in \mathcal{D}_{\text{valid}}$
    $a_d \leftarrow \text{round}(\tilde{w}_d \, n_{\text{lost}})$
    Adjust a_d (± 1) so $\sum_{d \in \mathcal{D}_{\text{valid}}} a_d = n_{\text{lost}}$
    $\hat{n}_d \leftarrow \hat{n}_d + a_d$ for all $d \in \mathcal{D}_{\text{valid}}$
$\mathcal{N} \leftarrow \varnothing$
for each $d \in \mathcal{D}_{\text{valid}}$ with $\hat{n}_d > 0$ **do**
    $i \leftarrow t^\star + d$
    Obtain token embeddings $\left(\phi(w^{(1)}), \ldots, \phi(w^{(L)})\right)$
    $\phi(w^{(i)}) \leftarrow \text{drop}\!\left(\phi(w^{(i)})\right)$
    Query the MLM to get $p\!\left(\mathcal{V}^{(i)} \mid x\right)$
    Compute $p_{\text{swap}}\!\left(w^{(i)}, \tilde{w}\right)$ for all $\tilde{w} \in \mathcal{V}^{(i)}$
    $\mathcal{S} \leftarrow$ top-$\hat{n}_d$ tokens by p_{swap}
    for each $\tilde{w} \in \mathcal{S}$ **do**
        $\tilde{x} \leftarrow x$ with $w^{(i)} \mapsto \tilde{w}$; $\mathcal{N} \leftarrow \mathcal{N} \cup \{\tilde{x}\}$
return $\mathcal{N}$ ▷ exactly N_{tot} neighbours, no padding

3.5 Experimental set-up

For comparability, the datasets and models used for the experiments are the same as in the CAMIA paper. The datasets are excerpts of six subsets of The Pile (19), made available by and streamed from the MIMIR repository (15). For each experiment, the following hyperparameters were used:

Table 3.1: Hyperparameter search space used in every experiment

Hyperparameter	**Values tested**
Number of neighbours n	{0, 25, 100}
Neighbourhood-signal weight in Edgington test W	{0, 9, 18, 36, 72}
Neighbourhood-signal feature importance in LR + PCA γ	{0, 1, 2, 3, 4}

Each test starts with $n = 0$ (and thus $W = 0,\ \gamma = 0$) to measure the baseline

CAMIA implementation for comparison. For $n \in \{25, 100\}$, the neighbourhood signal weight W and the feature-importance scale γ are varied pairwise but evaluated independently (since they belong to different tests). Hence nine hyperparameter configurations are used per test, as reported in Table 3.2. To account for variance, each configuration is run 10 times with different randomized splits of the dataset described in Section 3.5.0.1. This follows the protocol in (11).

Table 3.2: Neighbour-signal hyper-parameter grid explored in every experiment

Configuration	1	2	3	4	5	6	7	8	9
K (neighbours)	0	25	25	25	25	100	100	100	100
W (p-value weight)	0	9	18	36	72	9	18	36	72
Feature importance γ	0	1	2	3	4	1	2	3	4

The W levels are chosen with the baseline CAMIA signal count (72; see Table 2.1) in mind. The neighbourhood signal constitutes a standalone attack. It was therefore tested whether it merits disproportionate influence in the combined Edgington-style test. An analogous scaling is applied in the logistic-regression setting, where the neighbourhood block (after groupwise PCA) is multiplied by γ. The number of logistic-regression inputs equals the number of PCA components per feature family. In all experiments this is fixed to two components per family (as in CAMIA), yielding 20 PCA components from the 10 families plus the neighbourhood block.

3.5.0.1 Datasets

All six evaluation sets are components of The Pile (19), a 22-subset corpus for training and benchmarking language models. They were accessed through the MIMIR stream used in (11), as the original large release has since been withdrawn. Only excerpts could be streamed and processed. Table 3.3 briefly characterizes each component and reports its effective size within The Pile.

Table 3.3: Subsets of The Pile dataset used for experiments

Component	**Brief characterization**	**Effective size**
Pile-CC	Filtered Common Crawl subset with improved text extraction quality.	227.12 GiB
PubMed Central	Open-access full-text biomedical research articles from NCBI.	180.55 GiB
ArXiv	LaTeX sources for research preprints (predominantly math/CS/physics).	112.42 GiB
GitHub	Large corpus of open-source code repositories.	95.16 GiB
DM Mathematics	Natural-language math problems across algebra, calculus, etc.	15.49 GiB
HackerNews	Comment trees from the Y Combinator link aggregator.	7.80 GiB

Table 3.4: Composition of the evaluation datasets

Dataset	Split	Members	Non-members
ArXiv	`ngram_7_0.2`	500	500
DM Mathematics	`ngram_7_0.2`	89	89
GitHub	`ngram_7_0.2`	268	268
PubMed Central	`ngram_7_0.2`	491	491
HackerNews	`ngram_7_0.2`	646	646
Pile-CC	`ngram_7_0.2`	1 000	1 000

All runs follow the data-access protocol in Sections 2.2.2.2 and 2.2.2.3. For the non-member–only setting (CAMIA Edgington), $\alpha = 30\%$ of available non-members are drawn to construct a calibration subset. An empirical cumulative distribution function CDF_{D_f} is then computed per signal $f \in \mathcal{F}$ from these calibration scores. The target set consists of the remaining non-members together with all members. AUC and TPR@1%FPR are computed exclusively on the target set, with no overlap between calibration and target samples.

For the labelled setting (CAMIA LR + PCA), $\alpha = 30\%$ of members and $\alpha = 30\%$ of non-members are sampled to form D_{attack}. The per-group PCA projections and the logistic model are fit on D_{attack}, and the disjoint remainder is used as the target set. Each hyperparameter configuration is repeated across 10 independent runs with different random splits, as summarized in Tables 3.1–3.2.

Each row in Table 3.4 lists the number of member and non-member samples actually streamed and processed. Counts differ across datasets and are relatively small in DM Mathematics and GitHub, limiting statistical power. The member/non-member split is always balanced. A small pool of neighbours for both classes exists in the original MIMIR release. Because its generation procedure is unclear, (standard) neighbours were generated via the method in (16) (see Algorithm 1) for experiments 1 and 3 (see section 3.6).

3.5.0.2 Models and hardware

Table 3.5: Evaluated language-model checkpoints

Suite	Available checkpoints	Evaluated in this work
Pythia-deduped	70M, 160M, 1.4B, 2.8B, 6.9B, 12B	70M, 160M, 1.4B
Pythia original	70M, 160M, 1.4B, 2.8B, 6.9B, 12B	— (not run)
GPT-Neo	125M, 1.3B, 2.7B	125M, 1.3B

Due to time constraints, the 2.7B/2.8B/6.9B/12B models were not evaluated. Pythia-deduped variants were used (rather than the original Pythia checkpoints) to match how results are reported in (11).

All experiments were executed on AI Sweden's EdgeLab infrastructure using a VM with an NVIDIA Tesla T4 (16 GB VRAM) and 100 GB persistent storage, accessed

over VPN. Runs were conducted in Jupyter notebooks launched via Visual Studio Code's Remote SSH extension.

3.6 Attack variants evaluated

Table 3.6 summarizes the five experiments conducted for this thesis.

Table 3.6: Evaluated attack configurations

#	**New signals used**	**Neighbour type**
1	Neighbour	Standard
2	Neighbour	K highest-loss
3	Neighbour, Elbow, Flatline	Standard
4	Neighbour, Elbow, Flatline	Elbow-optimized
5	Neighbour, Elbow, Flatline	Flatline-optimized

Note that results corresponding to the custom implementation of the baseline CAMIA attack can be retrieved from the first two experiments with the hyperparameter setting $\{n = 0, W = 0, \gamma = 0\}$, since the neighbourhood signal is not used in that case. The neighbour signal is the novel signal incorporated in the CAMIA architecture based on the Neighbourhood Comparison Attack (16). Standard neighbours refers to the neighbours generated with Algorithm 1. K highest-loss neighbours refer to the neighbours generated with Algorithm 2. Elbow and flatline signals refer to the novel proposed signals described in sections 3.4.2 and 3.4.3. Elbow and flatline neighbours refer to the neighbours generated with Algorithm 3, using optimal elbow and flatline signal indices respectively.

4

Results

4.1 Experiment results

This chapter reports results from five experiments evaluating the attack configurations in Table 3.6. Each experiment follows the methodology in the previous chapter and is described in its corresponding section below.

For reference, results from the original CAMIA paper (11) are reproduced in Appendix A (see Section A.1). CAMIA was re-implemented to enable direct comparison. The custom implementation generally matches the original and is often close in absolute values, though numerical differences occur due to variance and implementation details. For brevity, full baseline replications are omitted, but they can be partially inferred from Tables 4.1, 4.2, 4.5, and 4.6 under the hyperparameter setting $\{n = 0, W = 0, \gamma = 0\}$. These differences do not affect the research questions: (1) whether loss-guided, targeted perturbations outperform random masking for membership inference attacks with semantics-preserving text edits (compare experiments 1 and 3 with 2, 4, and 5), and (2) whether a unified attack that combines and extends recent MIAs improves upon the state-of-the-art (compare all experiments against the original CAMIA results).

In the result tables that follow, for each dataset–model pair only the best score over the hyperparameter grid in Table 3.2 is reported. Each entry is a tuple $\{x, K, W/\gamma\}$ where x is AUC-ROC or TPR@1%FPR (determined by column), K is the number of neighbours, and W or γ is the weight or feature-importance of the neighbour signal. W is used for the CAMIA Edgington/Fisher/Pearson/George tests (Section 2.2.2.2); γ is used for CAMIA Logistic Regression + Group PCA (Section 2.2.2.3). Scores are averages over 10 runs with different randomized calibration/test splits (Section 3.5.0.1). The best result in each block is shown in **bold**.

4.1.1 Experiment 1: CAMIA with neighbour signal with standard neighbours

Experiment 1 is designed to test whether adding the Neighbourhood Comparison Attack (NCA) as a signal in the CAMIA architecture improves upon the results of the standard CAMIA attack. The neighbours used are the standard ones generated with Algorithm 1, as described in (16). Tables 4.1 and 4.2 report the result and best hyperparameter configuration for each combination of model, attack, dataset and metric, in a tuple of the form $\{x, K, W/\gamma\}$, where x is AUC-ROC or TPR@1%FPR (determined by column), K is the number of neighbours, and W or γ is the weight or feature-importance of the neighbour signal. If a tuple is of the form $\{x, 0, 0\}$, it means that the NCA with standard neighbours (Algorithm 1) did not improve upon the custom implementation of baseline CAMIA. If a tuple is of the form $\{x, y, z\}$, where $y, z \neq 0$, then NCA did improve results. The tuple corresponding to the best type of attack for each model and dataset is marked in **bold** in each column.

Table 4.1: Experiment 1 results on ArXiv, DM Mathematics and GitHub

Model	Attack	ArXiv		Mathematics		GitHub	
		AUC	TPR	AUC	TPR	AUC	TPR
P.D. 70M	Edgington	{76,0,0}	{15.1,0,0}	{93.4,0,0}	{50.1,0,0}	{85.4,100,18}	**{51.3,25,36}**
	Fisher	{27.2,25,72}	{1.1,25,18}	{9.6,25,72}	{1.1,0,0}	{16.5,25,72}	{0.3,25,18}
	Pearson	{73.6,25,9}	{14.3,0,0}	{90.8,0,0}	{31.6,0,0}	{85.3,100,18}	{46.9,25,72}
	George	**{76.3,100,9}**	**{21.2,0,0}**	**{95,0,0}**	**{75.6,0,0}**	{85.3,0,0}	{49,100,18}
	LR + PCA	**{76.3,0,0}**	{16.6,25,1}	{93.4,0,0}	{64.4,0,0}	**{85.9,0,0}**	{4.8,0,0}
P.D. 160M	Edgington	{78.2,0,0}	{20.1,0,0}	{90.9,0,0}	{28.8,0,0}	**{86.5,25,9}**	**{56.1,25,18}**
	Fisher	{25.7,25,72}	{0.9,25,18}	{9.5,100,72}	{0,0,0}	{15.8,25,72}	{0.3,25,9}
	Pearson	{76.3,0,0}	{14.5,0,0}	{87.5,0,0}	{14.4,0,0}	{86.2,25,9}	{54.3,25,36}
	George	**{78.4,0,0}**	{20.3,0,0}	**{95,0,0}**	{62.2,0,0}	**{86.5,0,0}**	{54.1,100,18}
	LR + PCA	**{78.4,0,0}**	**{26,100,4}**	{94.3,100,1}	**{68.1,100,1}**	{86.4,0,0}	{44.4,100,1}
P.D. 1.4B	Edgington	{80.4,0,0}	{23.7,0,0}	{82.5,0,0}	{7.8,0,0}	{89.1,100,9}	{58.9,25,9}
	Fisher	{22,100,72}	{0.4,100,36}	{14.9,100,72}	{0,0,0}	{13.9,25,72}	{0.3,25,18}
	Pearson	{78.5,0,0}	{18.5,25,9}	{75.7,0,0}	{2.9,0,0}	{88.7,100,9}	{49.9,25,18}
	George	**{80.5,25,9}**	{28,25,9}	{90.6,0,0}	{18.1,0,0}	**{89.1,0,0}**	**{59.7,100,9}**
	LR + PCA	{80.2,0,0}	**{32,25,1}**	**{92.7,25,1}**	**{42.4,25,4}**	{88.4,0,0}	{34.4,0,0}
G.N. 125M	Edgington	{80.1,0,0}	**{29.7,0,0}**	{91.3,0,0}	{31.1,0,0}	{86.9,100,9}	{52.4,25,18}
	Fisher	{22.8,100,72}	{0.4,0,0}	{10.4,100,72}	{0,0,0}	{15,25,72}	{0,0,0}
	Pearson	{77.8,25,9}	{25.7,0,0}	{86.7,0,0}	{18.7,0,0}	{86.5,100,18}	{49.8,100,36}
	George	**{80.4,25,9}**	{25.2,0,0}	**{95,0,0}**	**{60.5,0,0}**	{87.2,0,0}	**{56.7,25,18}**
	LR + PCA	{79.8,0,0}	{26,0,0}	{92.8,25,1}	{58.4,25,2}	**{87.7,0,0}**	{55.9,25,2}
G.N. 1.3B	Edgington	{81.4,0,0}	{30,0,0}	{86.7,0,0}	{21,0,0}	{89.4,100,9}	{60,100,9}
	Fisher	{21.1,25,72}	{0,0,0}	{15.5,100,72}	{0,0,0}	{13,25,72}	{0,0,0}
	Pearson	{80,100,9}	{24,100,9}	{80,0,0}	{10,0,0}	{88.7,100,9}	{50,100,9}
	George	**{81.5,100,9}**	{31,25,9}	**{92.5,0,0}**	**{61,0,0}**	**{89.6,0,0}**	**{64,25,18}**
	LR + PCA	{81.1,0,0}	**{32,0,0}**	{92,0,0}	{43,100,1}	{89.3,0,0}	**{64,0,0}**

Table 4.2: Experiment 1 results on PubMed Central, HackerNews and Pile-CC

Model	Attack	PubMed		HackerNews		Pile-CC	
		AUC	TPR	AUC	TPR	AUC	TPR
P.D. 70M	Edgington	{81.1,100,9}	{20.4,100,9}	{57.9,0,0}	{4.6,25,9}	{54.3,0,0}	**{4.0,100,18}**
	Fisher	{24.0,25,72}	{0.3,25,72}	{43.4,100,72}	{1.1,25,72}	{47.6,25,72}	{1.4,100,72}
	Pearson	{78.8,0,0}	{17.9,25,9}	{57.2,25,9}	{3.4,0,0}	**{54.4,100,18}**	{3.9,25,9}
	George	{81.3,0,0}	**{28.0,25,9}**	**{58.1,25,9}**	**{5.1,0,0}**	{54.3,0,0}	{3.5,100,72}
	LR + PCA	**{81.5,0,0}**	{25.6,0,0}	{55.9,100,2}	{3.1,25,1}	{52.1,0,0}	{1.2,0,0}
P.D. 160M	Edgington	{81.3,0,0}	{23.0,100,9}	{58.5,0,0}	{3.6,0,0}	{54.7,0,0}	**{4.8,100,72}**
	Fisher	{24.0,100,72}	{0.3,25,72}	{44.1,100,72}	{0.9,100,72}	{47.0,25,72}	{1.7,25,72}
	Pearson	{79.2,0,0}	{18.1,0,0}	{57.9,0,0}	{2.9,0,0}	{54.7,25,9}	{3.8,25,9}
	George	{81.9,0,0}	**{30.0,100,18}**	**{58.7,0,0}**	**{6.0,0,0}**	**{54.8,0,0}**	{3.7,25,36}
	LR + PCA	**{82.3,100,1}**	{28.9,0,0}	{56.9,100,2}	{4.2,0,0}	{53.5,0,0}	{2.0,25,3}
P.D. 1.4B	Edgington	{79.4,0,0}	{14.0,0,0}	{59.6,0,0}	{5.7,0,0}	{55.8,25,18}	{5.8,100,72}
	Fisher	{27.1,25,72}	{0.1,25,9}	{43.9,100,72}	{0.8,25,9}	{44.3,100,72}	{1.4,25,36}
	Pearson	{77.8,0,0}	{11.7,0,0}	{58.9,0,0}	{5.9,0,0}	{55.6,25,18}	{4.1,100,72}
	George	{79.7,0,0}	**{22.0,0,0}**	**{59.8,0,0}**	**{6.1,0,0}**	**{55.9,25,9}**	**{6.2,100,18}**
	LR + PCA	**{80.7,100,2}**	{21.0,25,3}	{57.5,100,4}	{4.2,0,0}	{55.2,0,0}	{4.2,100,2}
G.N. 125M	Edgington	**{83.1,25,9}**	{28.8,25,9}	**{58.1,0,0}**	**{4.0,100,9}**	{54.3,0,0}	{3.7,100,72}
	Fisher	{20.9,25,72}	{0.4,25,72}	{43.8,100,72}	{0.9,100,36}	{46.6,25,72}	{1.3,100,72}
	Pearson	{81.6,25,9}	{26.7,25,9}	{57.8,0,0}	{3.8,0,0}	{53.7,100,9}	{2.7,100,72}
	George	{83.0,0,0}	**{31.0,100,18}**	{58.0,0,0}	{3.9,25,9}	**{54.5,25,9}**	**{4.5,100,36}**
	LR + PCA	{82.3,0,0}	{23.5,100,1}	{57.8,100,2}	{2.5,25,1}	{52.8,0,0}	{2.0,0,0}
G.N. 1.3B	Edgington	{81.5,100,9}	{23.0,0,0}	**{59.3,0,0}**	{4.0,0,0}	**{55.1,25,9}**	{5.0,25,36}
	Fisher	{22.1,25,72}	{0.0,25,72}	{43.8,100,72}	{1.0,25,72}	{45.5,100,72}	{1.0,25,72}
	Pearson	{80.2,100,9}	{23.0,100,9}	{58.8,100,9}	**{5.0,0,0}**	{54.8,100,18}	{4.0,100,72}
	George	{81.5,0,0}	**{26.0,0,0}**	{59.2,0,0}	{4.0,0,0}	**{55.1,25,18}**	**{6.0,25,9}**
	LR + PCA	**{82.1,0,0}**	{21.0,0,0}	{56.0,100,3}	{3.0,0,0}	{54.2,0,0}	{3.0,0,0}

Section 4.2 contains a compilation of the best-performing experiment results. It can be checked whether the configuration in experiment 1 was optimal compared to the configuration of the other experiments, and if so whether it improved upon the results in the original CAMIA paper (11) in an absolute sense. Aggregate results of hyperparameter configurations are available in Section 4.2.6, and analyses and discussions of the results are available in Chapter 5.

4.1.2 Experiment 2: CAMIA with neighbour signal and K highest loss neighbours

Experiment 2 is designed to test whether adding the Neighbourhood Comparison Attack (NCA) as a signal in the CAMIA architecture improves upon the results of the standard CAMIA attack. The neighbours used are the K highest loss ones, generated with the new Algorithm 2, as described in 3.4. Tables 4.3 and 4.4 report the result and best hyperparameter configuration for each combination of model, attack, dataset and metric, in a tuple of the form $\{x, K, W/\gamma\}$, where x is AUC-ROC or TPR@1%FPR (determined by column), K is the number of neighbours, and W or γ is the weight or feature-importance of the neighbour signal. If a tuple is of the form $\{x, 0, 0\}$, it means that the NCA with K highest loss neighbours (Algorithm 2) did not improve upon the custom implementation of baseline CAMIA. If a tuple is of the form $\{x, y, z\}$, where $y, z \neq 0$, then NCA with the K highest loss neighbours did improve results. Experiment 2 also tells whether the custom neighbours improve upon the results from experiment 1 (see Section 4.1.1). The tuple corresponding to the best type of attack for each model and dataset is marked in **bold** in each column.

Table 4.3: Experiment 2 results on ArXiv, DM Mathematics and GitHub

Model	Attack	ArXiv		Mathematics		GitHub	
		AUC	TPR	AUC	TPR	AUC	TPR
P.D. 70M	Edgington	{76,0,0}	{15.1,0,0}	{93.4,0,0}	{50.1,0,0}	{85.4,25,18}	**{51.6,25,36}**
	Fisher	{27.2,25,72}	{1.0,25,36}	{9.5,25,72}	{1.1,0,0}	{16.5,100,72}	{0.3,25,18}
	Pearson	{73.6,100,9}	{14.3,0,0}	{90.8,0,0}	{31.6,0,0}	{85.3,25,18}	{47.6,25,72}
	George	**{76.3,25,9}**	**{21.2,0,0}**	**{95,0,0}**	**{75.6,0,0}**	{85.3,0,0}	{48.5,100,18}
	LR + PCA	**{76.3,0,0}**	{16.6,25,3}	{93.4,0,0}	{64.4,0,0}	**{85.9,0,0}**	{48.1,25,1}
P.D. 160M	Edgington	{78.2,0,0}	{20.1,0,0}	{90.9,0,0}	{28.8,0,0}	**{86.5,100,9}**	**{56.5,100,18}**
	Fisher	{25.6,25,72}	{0.9,25,36}	{9.5,25,72}	{0,0,0}	{15.9,25,72}	{0.3,25,9}
	Pearson	{76.3,0,0}	{14.5,0,0}	{87.5,0,0}	{14.4,0,0}	{86.2,25,9}	{54.2,100,36}
	George	**{78.4,0,0}**	{20.3,0,0}	**{95,0,0}**	{62.2,0,0}	**{86.5,0,0}**	{52.9,25,18}
	LR + PCA	**{78.4,0,0}**	**{25.9,25,3}**	{94.3,25,1}	**{68.4,25,1}**	{86.4,0,0}	{42.8,100,1}
P.D. 1.4B	Edgington	{80.4,0,0}	{24.0,0,0}	{82.5,0,0}	{8.0,0,0}	**{89.1,100,9}**	**{59.0,100,9}**
	Fisher	{21.9,25,72}	{1.0,100,72}	{15.1,100,72}	{0,0,0}	{14.0,100,72}	{0,25,18}
	Pearson	{78.6,25,9}	{19.0,100,9}	{75.7,0,0}	{3.0,0,0}	{88.6,100,9}	{50.0,100,18}
	George	**{80.6,25,9}**	{28.0,100,9}	{90.6,0,0}	{18.0,0,0}	**{89.1,0,0}**	**{59.0,25,9}**
	LR + PCA	{80.2,0,0}	**{32.0,25,1}**	**{92.8,25,1}**	**{43.0,25,3}**	{88.4,0,0}	{34.0,0,0}
G.N. 125M	Edgington	{80.1,0,0}	**{28.6,0,0}**	{91.2,0,0}	{32.2,0,0}	{87.0,25,9}	{51.9,25,9}
	Fisher	{23.0,100,72}	{0.4,0,0}	{11.0,25,72}	{0,0,0}	{15.0,100,72}	{0,0,0}
	Pearson	{78.1,100,9}	{22.6,0,0}	{86.8,0,0}	{21.4,0,0}	{86.6,100,18}	{47.1,100,36}
	George	**{80.3,25,9}**	{24.7,0,0}	**{94.7,0,0}**	{60.3,0,0}	{87.4,0,0}	{56.6,25,18}
	LR + PCA	{79.7,0,0}	{26.5,0,0}	{92.7,25,1}	**{60.8,25,1}**	**{88.0,0,0}**	**{62.1,25,1}**
G.N. 1.3B	Edgington	{81.4,0,0}	{30.0,0,0}	{86.7,0,0}	{21.0,0,0}	{89.4,100,9}	{60.0,100,9}
	Fisher	{21.0,25,72}	{0,0,0}	{15.5,25,72}	{1.0,100,72}	{13.0,100,72}	{0,0,0}
	Pearson	{80.1,25,9}	{24.0,25,9}	{80.0,0,0}	{10.0,0,0}	{88.7,100,9}	**{49.0,100,9}**
	George	**{81.5,25,9}**	{31.0,0,0}	**{92.5,0,0}**	**{61.0,0,0}**	**{89.6,0,0}**	**{64.0,100,18}**
	LR + PCA	{81.1,0,0}	**{32.0,0,0}**	{92.0,0,0}	{43.0,25,1}	{89.3,0,0}	**{64.0,0,0}**

Table 4.4: Experiment 2 results on PubMed Central, HackerNews and Pile-CC

Model	Attack	PubMed		HackerNews		Pile-CC	
		AUC	TPR	AUC	TPR	AUC	TPR
P.D. 70M	Edgington	{81,25,9}	{20.6,25,9}	{57.9,0,0}	{4.5,100,9}	{54.3,0,0}	{3.9,25,9}
	Fisher	{24.2,25,72}	{0.3,25,72}	{43.1,25,72}	{1.1,100,72}	{47.5,100,72}	{1.4,25,36}
	Pearson	{78.8,0,0}	{18.1,100,9}	{57.3,100,9}	{3.4,0,0}	**{54.4,100,18}**	**{4.0,25,9}**
	George	{81.3,0,0}	**{28.1,25,9}**	**{58.2,100,9}**	**{5.1,0,0}**	{54.3,0,0}	{3.4,100,72}
	LR + PCA	**{81.5,0,0}**	{25.6,0,0}	{55.8,25,3}	{3.1,0,0}	{52.1,0,0}	{1.2,0,0}
P.D. 160M	Edgington	{81.3,0,0}	{22.9,100,9}	{58.5,0,0}	{3.6,0,0}	{54.7,0,0}	**{4.7,0,0}**
	Fisher	{24.1,25,72}	{0.3,25,36}	{43.8,25,72}	{0.9,25,72}	{46.9,100,72}	{1.8,25,72}
	Pearson	{79.2,0,0}	{18.1,0,0}	{57.9,0,0}	{2.9,0,0}	{54.7,100,9}	{3.9,25,72}
	George	{81.9,0,0}	**{29.9,25,9}**	**{58.7,0,0}**	**{6.0,0,0}**	**{54.8,0,0}**	{3.5,25,36}
	LR + PCA	**{82.3,25,1}**	{28.9,0,0}	{56.7,25,4}	{4.2,0,0}	{53.5,0,0}	{2.0,0,0}
P.D. 1.4B	Edgington	{79.4,0,0}	{14.0,0,0}	{59.6,0,0}	**{6.0,0,0}**	**{55.9,100,18}**	**{6.0,100,72}**
	Fisher	{27.2,25,72}	{0,25,9}	{43.5,25,72}	{1.0,25,9}	{44.2,0,0}	{1.0,100,36}
	Pearson	{77.8,0,0}	{12.0,0,0}	{58.9,0,0}	**{6.0,0,0}**	{55.7,100,36}	{5.0,100,72}
	George	{79.7,0,0}	**{22.0,0,0}**	**{59.8,0,0}**	**{6.0,0,0}**	**{55.9,100,36}**	**{6.0,25,18}**
	LR + PCA	**{80.8,100,3}**	**{22.0,25,2}**	{57.4,25,3}	{4.0,0,0}	{55.2,0,0}	{4.0,25,3}
G.N. 125M	Edgington	**{82.9,100,9}**	{28.3,25,9}	{58.0,0,0}	**{4.1,0,0}**	{54.2,0,0}	{3.7,100,72}
	Fisher	{21.2,25,72}	{0.4,25,72}	{43.8,25,72}	{1.2,0,0}	{46.6,25,72}	{1.4,0,0}
	Pearson	{81.4,100,9}	{25.6,25,9}	{57.6,0,0}	{3.9,0,0}	{53.7,100,9}	{2.8,25,72}
	George	**{82.9,0,0}**	**{30.6,25,18}**	**{58.2,0,0}**	{3.8,100,9}	**{54.4,100,9}**	**{4.4,25,36}**
	LR + PCA	{82.7,0,0}	{24.8,100,1}	{57.4,25,2}	{2.5,100,1}	{52.8,0,0}	{2.0,0,0}
G.N. 1.3B	Edgington	{81.4,25,9}	{23.0,0,0}	**{59.3,0,0}**	{4.0,0,0}	{55.1,25,18}	{5.0,100,72}
	Fisher	{22.3,100,72}	{1.0,25,72}	{43.4,25,72}	{2.0,25,72}	{45.3,100,72}	{1.0,25,72}
	Pearson	{80.1,0,0}	{22.0,25,9}	{58.9,100,9}	**{5.0,0,0}**	{54.8,25,18}	{4.0,25,72}
	George	{81.5,0,0}	**{26.0,0,0}**	{59.2,0,0}	{4.0,0,0}	**{55.2,25,36}**	**{6.0,100,9}**
	LR + PCA	**{82.1,100,1}**	{21.0,0,0}	{55.9,25,3}	{3.0,0,0}	{54.2,0,0}	{3.0,0,0}

Section 4.2 contains a compilation of the best-performing experiment results. It can be checked whether the configuration in experiment 2 was optimal compared to the configuration of the other experiments, and if so whether it improved upon the results in the original CAMIA paper (11) in an absolute sense. Aggregate results of hyperparameter configurations are available in Section 4.2.6, and analyses and discussions of the results are available in Chapter 5.

4.1.3 Experiment 3: CAMIA with all new signals and standard neighbours

Experiment 3 is designed to test whether adding the Neighbourhood Comparison Attack (NCA), as well as the novel custom elbow and flatline signals described in Sections 3.4.2 and 3.4.3 as signals in the CAMIA architecture improves upon the results of the standard CAMIA attack. The neighbours used are the standard ones generated with Algorithm 1, as described in (16). Tables 4.5 and 4.6 report the result and best hyperparameter configuration for each combination of model, attack, dataset and metric, in a tuple of the form $\{x, K, W/\gamma\}$, where x is AUC-ROC or TPR@1%FPR (determined by column), K is the number of neighbours, and W or γ is the weight or feature-importance of the neighbour signal. If a tuple is of the form $\{x, 0, 0\}$, it means that the NCA with standard neighbours (Algorithm 1) and the elbow and flatline signals did not improve upon the custom implementation of baseline CAMIA. If a tuple is of the form $\{x, y, z\}$, where $y, z \neq 0$, then the NCA, elbow and flatline signals did improve results. Experiment 3 also tells whether including the elbow and flatline signals improves upon the results from experiment 1 (see Section 4.1.1). The tuple corresponding to the best type of attack for each model and dataset is marked in **bold** in each column.

Table 4.5: Experiment 3 results on ArXiv, DM Mathematics and GitHub

Model	Attack	ArXiv		Mathematics		GitHub	
		AUC	TPR	AUC	TPR	AUC	TPR
P.D. 70M	Edgington	{76.2,0,0}	{15.7,0,0}	{93.3,0,0}	{52.2,0,0}	{85.3,100,18}	**{51.4,100,18}**
	Fisher	{27.1,25,72}	{1.1,25,18}	{9.5,25,72}	{1.1,0,0}	{16.5,25,72}	{0.3,25,72}
	Pearson	{73.7,25,9}	{14.6,0,0}	{90.4,0,0}	{34.3,0,0}	{85.2,100,18}	{45.2,25,72}
	George	**{76.4,25,9}**	**{21.1,0,0}**	**{95,0,0}**	**{75.9,0,0}**	{85.1,0,0}	{48.6,100,18}
	LR + PCA	{75.8,0,0}	{16.2,0,0}	{93.5,0,0}	{62.9,0,0}	**{85.6,0,0}**	{47.2,0,0}
P.D. 160M	Edgington	{78.3,0,0}	{21.5,0,0}	{90.1,0,0}	{27.1,0,0}	**{86.5,25,9}**	**{56.0,100,18}**
	Fisher	{25.6,25,72}	{0.8,25,36}	{9.7,100,72}	{0,0,0}	{15.9,25,72}	{0.3,25,9}
	Pearson	{76.6,0,0}	{14.0,0,0}	{86.3,0,0}	{12.9,0,0}	{86.2,100,9}	{53.2,25,36}
	George	**{78.5,0,0}**	{20.6,0,0}	**{94.8,0,0}**	{62.2,0,0}	**{86.5,0,0}**	{53.7,100,18}
	LR + PCA	{77.9,0,0}	**{24.9,0,0}**	{94.5,25,1}	**{71.7,25,2}**	{85.8,0,0}	{46.7,100,4}
P.D. 1.4B	Edgington	{80.5,0,0}	{22.9,0,0}	{80.9,0,0}	{8.4,0,0}	**{89.1,100,9}**	{58.5,25,9}
	Fisher	{21.8,100,72}	{0.4,0,0}	{15.3,100,72}	{0,0,0}	{13.9,25,72}	{0.3,25,36}
	Pearson	{78.7,0,0}	{17.8,25,9}	{73.8,0,0}	{3.3,0,0}	{88.6,100,9}	{49.8,25,18}
	George	**{80.6,100,9}**	{28.1,100,9}	{90,0,0}	{18.7,0,0}	**{89.1,0,0}**	**{59.6,100,18}**
	LR + PCA	{79.8,0,0}	**{32,0,0}**	{94,25,2}	**{51.6,25,3}**	{88.5,0,0}	{32.8,0,0}
G.N. 125M	Edgington	{80.2,0,0}	**{28.6,0,0}**	{90.6,0,0}	{28.6,0,0}	{86.8,100,9}	{51.6,25,18}
	Fisher	{22.7,100,72}	{0.4,0,0}	{10.6,100,72}	{0,0,0}	{15.1,25,72}	{0,0,0}
	Pearson	{78.0,25,9}	{22.9,0,0}	{85.7,0,0}	{14.9,0,0}	{86.4,100,18}	{48.4,25,36}
	George	**{80.5,0,0}**	{25.7,0,0}	**{94.7,0,0}**	{59.7,0,0}	{87.1,0,0}	**{56.0,25,18}**
	LR + PCA	{79.4,0,0}	{26.0,0,0}	{92.4,0,0}	{57.5,25,4}	**{87.6,0,0}**	{53.5,25,1}
G.N. 1.3B	Edgington	{81.5,0,0}	**{31.8,0,0}**	{85.4,0,0}	{18.7,0,0}	{89.4,100,9}	{58.7,100,9}
	Fisher	{21.0,25,72}	{0.2,0,0}	{15.8,100,72}	{0,0,0}	{12.9,25,72}	{0,0,0}
	Pearson	{80.2,25,9}	{25.8,0,0}	{78.3,0,0}	{7.3,0,0}	{88.6,100,9}	{49.5,25,9}
	George	**{81.6,100,9}**	{31.3,25,9}	**{92.1,0,0}**	**{59.2,0,0}**	**{89.6,0,0}**	**{63.8,25,18}**
	LR + PCA	{80.6,0,0}	{30.8,0,0}	**{93.5,0,0}**	{57.0,0,0}	{88.9,0,0}	{57.2,0,0}

Table 4.6: Experiment 3 results on PubMed Central, HackerNews and Pile-CC

Model	Attack	PubMed		HackerNews		Pile-CC	
		AUC	TPR	AUC	TPR	AUC	TPR
P.D. 70M	Edgington	{81,100,9}	{20.2,100,9}	{58,0,0}	{4.3,25,9}	{54.3,0,0}	**{4,100,9}**
	Fisher	{24.1,25,72}	{0.3,25,72}	{43.2,100,72}	{1.1,25,36}	{47.6,25,72}	{1.4,100,72}
	Pearson	{78.6,0,0}	{16.9,100,9}	{57.3,0,0}	{3.3,25,9}	**{54.4,25,18}**	{3.6,25,9}
	George	{81.1,0,0}	**{28,100,9}**	**{58.3,25,9}**	**{5.2,0,0}**	{54.3,0,0}	{3.5,100,72}
	LR + PCA	**{81.2,0,0}**	{20.1,0,0}	{55.8,100,3}	{3.5,0,0}	{52,0,0}	{1.2,0,0}
P.D. 160M	Edgington	{81.2,0,0}	{21.5,25,9}	{58.6,0,0}	{3.6,0,0}	{54.8,0,0}	**{4.7,100,72}**
	Fisher	{24,100,72}	{0.3,25,72}	{4.4,100,72}	{0.8,25,72}	{47,25,72}	{1.6,25,72}
	Pearson	{79.2,0,0}	{15.6,0,0}	{58,0,0}	{2.5,0,0}	{54.7,100,9}	{3.9,25,9}
	George	**{81.8,0,0}**	**{29.3,100,18}**	**{58.8,0,0}**	**{5.9,0,0}**	**{54.9,0,0}**	{3.8,25,36}
	LR + PCA	{81.5,0,0}	{26,0,0}	{56.9,100,4}	{3.9,0,0}	{53.6,0,0}	{1.9,0,0}
P.D. 1.4B	Edgington	{79.3,0,0}	{12.3,0,0}	{59.7,0,0}	**{5.6,0,0}**	**{55.8,25,18}**	**{5.8,100,72}**
	Fisher	{27,25,72}	{0.1,0,0}	{43.8,25,72}	{0.8,25,9}	{44.3,100,72}	{1.4,100,72}
	Pearson	{77.7,0,0}	{10.6,0,0}	{58.9,0,0}	**{6,0,0}**	{55.6,25,36}	{4.2,100,36}
	George	{79.7,0,0}	**{21.9,0,0}**	**{59.9,0,0}**	**{6.1,0,0}**	**{55.8,25,36}**	**{6.1,25,18}**
	LR + PCA	**{80.6,100,3}**	{19.5,25,2}	{57.7,100,1}	{4.9,0,0}	{54.8,0,0}	{4.2,0,0}
G.N. 125M	Edgington	**{83,100,9}**	{27.4,25,9}	{58.2,0,0}	**{3.8,0,0}**	{54.3,0,0}	{3.7,100,72}
	Fisher	{20.9,25,72}	{0.4,25,72}	{43.7,25,72}	{0.7,0,0}	{46.6,25,72}	{1.4,0,0}
	Pearson	{81.5,25,9}	{25.6,25,9}	{57.8,0,0}	{3.3,100,9}	{53.7,100,9}	{2.5,100,72}
	George	**{82.9,0,0}**	**{31.1,100,18}**	**{58.2,0,0}**	{3.8,25,9}	**{54.5,100,9}**	**{4.4,25,36}**
	LR + PCA	{81.8,0,0}	{19.1,100,1}	{57.1,25,2}	{2.5,100,1}	{53.1,0,0}	{2,0,0}
G.N. 1.3B	Edgington	**{81.4,100,9}**	{19.9,100,9}	**{59.4,0,0}**	{4.7,0,0}	**{55.1,25,9}**	{5.3,100,18}
	Fisher	{22.1,25,72}	{0.3,25,72}	{43.6,25,72}	{1.3,25,72}	{45.5,100,72}	{1,25,72}
	Pearson	{80.1,0,0}	{21.7,25,9}	{58.8,100,9}	**{5.5,0,0}**	{54.7,25,18}	{4.2,25,72}
	George	{81.4,0,0}	**{25.4,0,0}**	{59.4,0,0}	{4,25,18}	**{55.1,25,36}**	**{5.7,100,9}**
	LR + PCA	**{81.4,0,0}**	{19.4,0,0}	{55.7,25,3}	{3.4,25,1}	{54.5,0,0}	{3.4,0,0}

Section 4.2 contains a compilation of the best-performing experiment results. It can be checked whether the configuration in experiment 3 was optimal compared to the configuration of the other experiments, and if so whether it improved upon the results in the original CAMIA paper (11) in an absolute sense. Aggregate results of hyperparameter configurations are available in Section 4.2.6, and analyses and discussions of the results are available in Chapter 5.

4.1.4 Experiment 4: CAMIA with all new signals and elbow neighbours

Experiment 4 is designed to test whether adding the Neighbourhood Comparison Attack (NCA), as well as the novel custom elbow and flatline signals described in Sections 3.4.2 and 3.4.3 as signals in the CAMIA architecture improves upon the results of the standard CAMIA attack. Elbow neighbours generated with Algorithm 3 are used, with central perturbation index identified through the procedure described in 3.4.2. Tables 4.7 and 4.8 report the result and best hyperparameter configuration for each combination of model, attack, dataset and metric, in a tuple of the form $\{x, K, W/\gamma\}$, where x is AUC-ROC or TPR@1%FPR (determined by column), K is the number of neighbours, and W or γ is the weight or feature-importance of the neighbour signal. If a tuple is of the form $\{x, 0, 0\}$, it means that the NCA with elbow neighbours (Algorithm 3) and the elbow and flatline signals did not improve upon the custom implementation of baseline CAMIA. If a tuple is of the form $\{x, y, z\}$, where $y, z \neq 0$, then the NCA with elbow neighbours as well as elbow and flatline signals did improve results. Experiment 4 also tells whether including the elbow neighbours improves upon the results from experiment 3, where standard neighbours are used (see Section 4.1.3). The tuple corresponding to the best type of attack for each model and dataset is marked in **bold** in each column.

Table 4.7: Experiment 4 results on ArXiv, DM Mathematics and GitHub

Model	Attack	ArXiv		Mathematics		GitHub	
		AUC	TPR	AUC	TPR	AUC	TPR
P.D. 70M	Edgington	{75.8,0,0}	{15.8,0,0}	{92.9,0,0}	{54.4,0,0}	{85.2,100,18}	**{51.1,100,36}**
	Fisher	{27.7,25,72}	{0.9,25,72}	{9.4,25,72}	{0.3,0,0}	{16.6,25,72}	{0.3,25,9}
	Pearson	{73.8,0,0}	{13.3,0,0}	{90.5,0,0}	{30.0,0,0}	{85.4,25,9}	{48.1,25,72}
	George	**{76.1,0,0}**	**{21.3,0,0}**	**{95,0,0}**	**{77.5,0,0}**	{85.2,0,0}	{46.9,25,18}
	LR + PCA	{75.2,0,0}	{15.8,0,0}	{93.6,0,0}	{60.5,0,0}	**{85.5,0,0}**	{49.5,25,1}
P.D. 160M	Edgington	{78.3,0,0}	{20.9,0,0}	{90.1,0,0}	{27.1,0,0}	{86.5,100,9}	**{56.5,100,18}**
	Fisher	{25.6,100,72}	{0.9,100,36}	{9.7,100,72}	{0,0,0}	{15.9,25,72}	{0.3,25,9}
	Pearson	{76.6,0,0}	{14.0,100,18}	{86.3,0,0}	{12.9,0,0}	{86.1,25,9}	{52.9,100,36}
	George	**{78.5,0,0}**	{20.6,0,0}	**{94.8,0,0}**	{62.2,0,0}	**{86.5,0,0}**	{52.8,25,18}
	LR + PCA	{77.8,0,0}	**{25.1,0,0}**	{94.6,100,1}	**{71.9,100,2}**	{86,0,0}	{45.7,25,1}
P.D. 1.4B	Edgington	{80.5,0,0}	{22.9,0,0}	{80.9,0,0}	{8.4,0,0}	**{89.1,25,9}**	{58.4,25,9}
	Fisher	{21.8,25,72}	{0.7,100,72}	{15.6,100,72}	{0,0,0}	{13.9,25,72}	{0.3,25,36}
	Pearson	{78.8,25,9}	{18.1,100,9}	{73.8,0,0}	{3.3,0,0}	{88.6,100,9}	{49.3,25,18}
	George	**{80.7,100,9}**	{28.1,0,0}	{90,0,0}	{18.7,0,0}	**{89.1,0,0}**	**{58.9,25,9}**
	LR + PCA	{79.8,0,0}	**{32,25,4}**	**{94.1,100,1}**	**{51.7,25,3}**	{88.5,0,0}	{32.8,0,0}
G.N. 125M	Edgington	{80.2,0,0}	**{29.5,0,0}**	{90.5,0,0}	{29.4,0,0}	{86.9,25,9}	{51,25,9}
	Fisher	{22.9,100,72}	{0.4,0,0}	{10.8,25,72}	{0,0,0}	{15.1,100,72}	{0,0,0}
	Pearson	{78.3,25,9}	{21.9,0,0}	{85.7,0,0}	{17.6,0,0}	{86.5,25,18}	{44.1,100,36}
	George	**{80.4,100,9}**	{25.2,0,0}	**{94.4,0,0}**	{59.5,0,0}	{87.3,0,0}	**{56.1,25,18}**
	LR + PCA	{79.2,0,0}	{26.1,0,0}	{92.5,25,1}	{57.1,25,1}	**{88,0,0}**	**{59.8,100,1}**
G.N. 1.3B	Edgington	{81.5,0,0}	**{31.8,0,0}**	{85.4,0,0}	{18.7,0,0}	{89.4,25,9}	{58.7,25,9}
	Fisher	{21,25,72}	{0.2,0,0}	{16,100,72}	{0,0,0}	{13,25,72}	{0,0,0}
	Pearson	{80.2,25,9}	{25.8,0,0}	{78.3,0,0}	{7.3,0,0}	{88.6,25,9}	{49.5,25,9}
	George	**{81.6,0,0}**	{31.3,0,0}	**{92.1,0,0}**	**{59.2,0,0}**	**{89.6,0,0}**	**{63.7,25,18}**
	LR + PCA	{80.6,0,0}	{30.8,25,3}	**{93.5,0,0}**	{57,0,0}	{88.9,0,0}	{57.2,0,0}

Table 4.8: Experiment 4 results on PubMed Central, HackerNews and Pile-CC

Model	Attack	PubMed		HackerNews		Pile-CC	
		AUC	TPR	AUC	TPR	AUC	TPR
P.D. 70M	Edgington	{81,100,9}	{17.4,25,9}	{58.4,0,0}	{4.2,25,18}	{54.2,0,0}	**{4.1,25,72}**
	Fisher	{24.4,25,72}	{0.3,25,72}	{42.9,25,72}	{0.8,25,72}	{47.7,25,72}	{1.5,100,36}
	Pearson	{79.7,0,0}	{16.7,25,9}	{57.8,0,0}	{3.2,100,36}	**{54.3,100,18}**	{3.5,25,36}
	George	**{81.5,0,0}**	**{28.9,100,9}**	**{58.6,0,0}**	**{5.3,0,0}**	{54.1,0,0}	{3.4,25,36}
	LR + PCA	{81,0,0}	{21.5,0,0}	{55.1,25,3}	{3.6,0,0}	{52.2,0,0}	{1.3,0,0}
P.D. 160M	Edgington	{81.2,0,0}	{21.5,25,9}	{58.6,0,0}	{3.6,0,0}	{54.8,0,0}	**{4.5,25,72}**
	Fisher	{24.4,25,72}	{0.3,25,36}	{43.6,100,72}	{0.7,25,72}	{47,25,72}	{1.7,25,36}
	Pearson	{79.2,0,0}	{17.4,25,18}	{58.1,0,0}	{2.5,0,0}	{54.7,100,9}	{3.8,25,18}
	George	**{81.8,0,0}**	**{29,100,9}**	**{58.9,0,0}**	**{5.9,0,0}**	**{54.9,0,0}**	{3.6,100,36}
	LR + PCA	{81.6,25,1}	{25.7,0,0}	{56.7,100,3}	{3.8,0,0}	{53.7,0,0}	{1.9,0,0}
P.D. 1.4B	Edgington	{79.3,0,0}	{12.3,0,0}	{59.7,0,0}	**{5.6,0,0}**	**{55.9,100,18}**	**{6.3,100,72}**
	Fisher	{27.2,25,72}	{0.2,25,36}	{43.2,25,72}	{0.8,100,9}	{44.3,0,0}	{1.3,100,72}
	Pearson	{77.8,0,0}	{10.6,0,0}	{58.9,0,0}	**{6.2,0,0}**	{55.7,100,36}	{4.9,100,72}
	George	{79.7,0,0}	**{21.9,0,0}**	**{59.9,0,0}**	{6.1,0,0}	**{55.9,100,18}**	{6.2,100,18}
	LR + PCA	**{80.7,25,2}**	{20.7,25,2}	{57.7,0,0}	{4.9,0,0}	{54.9,0,0}	{4.3,0,0}
G.N. 125M	Edgington	**{82.8,100,9}**	{27.3,25,9}	**{58.1,0,0}**	**{4.1,0,0}**	{54.2,0,0}	{3.8,100,72}
	Fisher	{21.4,25,72}	{0.3,25,72}	{43.7,100,72}	{1.0,0,0}	{46.6,100,72}	{1.5,25,9}
	Pearson	{81.3,25,9}	{25.1,25,9}	{57.6,0,0}	{4,0,0}	{53.7,100,9}	{2.5,25,72}
	George	{82.8,0,0}	**{30.7,25,18}**	{57.9,0,0}	{3.8,100,9}	**{54.4,0,0}**	**{4.5,100,36}**
	LR + PCA	{82.1,0,0}	{21,0,0}	{56.8,100,4}	{2.3,0,0}	{53,0,0}	{2,25,2}
G.N. 1.3B	Edgington	{81.3,100,9}	{19.9,0,0}	**{59.4,0,0}**	{4.7,0,0}	**{55.1,25,9}**	{5.5,100,72}
	Fisher	{22.3,25,72}	{0.5,25,72}	{4.3,100,72}	{1.4,25,72}	{45.5,25,72}	{0.9,25,72}
	Pearson	{80,25,9}	{21.2,25,9}	{58.9,25,9}	**{5.5,0,0}**	{54.7,100,18}	{4.7,100,72}
	George	{81.4,0,0}	**{25.4,0,0}**	**{59.4,0,0}**	{4,0,0}	**{55.1,100,18}**	**{5.9,100,9}**
	LR + PCA	**{81.5,100,1}**	{19.4,0,0}	{55.6,100,4}	{3.4,25,1}	{54.5,0,0}	{3.4,0,0}

Section 4.2 contains a compilation of the best-performing experiment results. It can be checked whether the configuration in experiment 4 was optimal compared to the configuration of the other experiments, and if so whether it improved upon the results in the original CAMIA paper (11) in an absolute sense. Aggregate results of hyperparameter configurations are available in Section 4.2.6, and analyses and discussions of the results are available in Chapter 5.

4.1.5 Experiment 5: CAMIA with all new signals and flatline neighbours

Experiment 5 is designed to test whether adding the Neighbourhood Comparison Attack (NCA), as well as the novel custom elbow and flatline signals described in Sections 3.4.2 and 3.4.3 as signals in the CAMIA architecture improves upon the results of the standard CAMIA attack. Flatline neighbours generated with Algorithm 3 are used, with central perturbation index identified through the procedure described in 3.4.3. Tables 4.9 and 4.10 report the result and best hyperparameter configuration for each combination of model, attack, dataset and metric, in a tuple of the form $\{x, K, W/\gamma\}$, where x is AUC-ROC or TPR@1%FPR (determined by column), K is the number of neighbours, and W or γ is the weight or feature-importance of the neighbour signal. If a tuple is of the form $\{x, 0, 0\}$, it means that the NCA with flatline neighbours (Algorithm 3) and the elbow and flatline signals did not improve upon the custom implementation of baseline CAMIA. If a tuple is of the form $\{x, y, z\}$, where $y, z \neq 0$, then the NCA with flatline neighbours as well as elbow and flatline signals did improve results. Experiment 5 also tells whether including the elbow neighbours improves upon the results from experiment 3, where standard neighbours are used (see Section 4.1.3). It also shows whether results improve compared to when elbow neighbours are used in experiment 4 (see Section 4.1.4). The tuple corresponding to the best type of attack for each model and dataset is marked in **bold** in each column.

Table 4.9: Experiment 5 results on ArXiv, DM Mathematics and GitHub

Model	Attack	ArXiv		Mathematics		GitHub	
		AUC	TPR	AUC	TPR	AUC	TPR
P.D. 70M	Edgington	{75.8,0,0}	{15.8,0,0}	{92.9,0,0}	{54.4,0,0}	{85.3,25,18}	**{51.1,25,36}**
	Fisher	{27.7,100,72}	{0.8,25,18}	{9.6,100,72}	{0.3,0,0}	{16.6,100,72}	{0.3,25,9}
	Pearson	{73.8,0,0}	{13.3,0,0}	{90.5,0,0}	{30,0,0}	**{85.4,100,9}**	{49.1,25,72}
	George	**{76.1,0,0}**	**{21.3,0,0}**	**{95,0,0}**	**{77.5,0,0}**	{85.2,0,0}	{46.9,25,18}
	LR + PCA	{75.2,0,0}	{15.8,0,0}	{93.6,0,0}	{60.5,0,0}	{85.5,0,0}	{49.5,25,1}
P.D. 160M	Edgington	{78.3,0,0}	{20.9,0,0}	{90.1,0,0}	{27.1,0,0}	{86.5,25,9}	**{56.6,25,18}**
	Fisher	{25.5,100,72}	{0.8,25,36}	{9.7,100,72}	{0,0,0}	{15.9,100,72}	{0.3,25,9}
	Pearson	{76.6,0,0}	{13.9,100,18}	{86.3,0,0}	{12.9,0,0}	{86.1,25,9}	{53.4,25,36}
	George	**{78.5,0,0}**	{20.6,0,0}	**{94.8,0,0}**	{62.2,0,0}	**{86.5,0,0}**	{53,100,18}
	LR + PCA	{77.8,0,0}	**{25.1,0,0}**	{94.5,100,1}	**{71.7,25,2}**	{86,0,0}	{45.7,25,1}
P.D. 1.4B	Edgington	{80.5,0,0}	{23.3,25,9}	{80.9,0,0}	{8.4,0,0}	**{89.1,25,9}**	{57.3,25,9}
	Fisher	{21.9,100,72}	{0.6,100,72}	{15.6,25,72}	{0,0,0}	{14,25,72}	{0.3,25,36}
	Pearson	{78.8,25,9}	{18.1,25,9}	{73.8,0,0}	{3.3,0,0}	{88.6,100,9}	{47.9,100,18}
	George	**{80.7,25,9}**	{28.1,0,0}	{90,0,0}	{18.7,0,0}	**{89.1,0,0}**	**{58.9,100,9}**
	LR + PCA	{79.8,0,0}	**{32.1,25,3}**	**{94,100,1}**	**{51.6,100,4}**	{88.5,0,0}	{32.8,0,0}
G.N. 125M	Edgington	**{80.2,0,0}**	**{28.7,0,0}**	{90.6,0,0}	{28.6,0,0}	**{86.8,100,9}**	{51.4,100,18}
	Fisher	{22.7,100,72}	{0.4,0,0}	{10.9,100,72}	{0,0,0}	{15.1,25,72}	{0,0,0}
	Pearson	{78,25,9}	{22.9,0,0}	{85.7,0,0}	{14.9,0,0}	{86.4,100,18}	{48.5,25,36}
	George	**{80.5,0,0}**	{25.7,0,0}	**{94.7,0,0}**	{59.7,0,0}	{87.1,0,0}	**{56.2,100,9}**
	LR + PCA	{79.5,0,0}	{25.8,0,0}	{92.4,0,0}	{57.8,25,2}	{87.6,0,0}	{51.8,25,1}
G.N. 1.3B	Edgington	**{81.5,0,0}**	**{31.6,0,0}**	{85.4,0,0}	{18.7,0,0}	{89.3,100,9}	{58.1,100,9}
	Fisher	{20.9,100,72}	{0.2,0,0}	{15.5,100,72}	{0,0,0}	{13,25,72}	{0,0,0}
	Pearson	{80.2,100,9}	{24,0,0}	{78.3,0,0}	{7.3,0,0}	{88.6,100,9}	{48.4,25,9}
	George	**{81.6,0,0}**	**{31.3,0,0}**	**{92.1,0,0}**	{59.2,0,0}	**{89.6,0,0}**	**{63.9,25,18}**
	LR + PCA	{80.6,0,0}	{30.8,25,2}	{93.5,0,0}	{57,0,0}	{88.9,0,0}	{57.2,0,0}

Table 4.10: Experiment 5 results on PubMed Central, HackerNews and Pile-CC

Model	Attack	PubMed		HackerNews		Pile-CC	
		AUC	TPR	AUC	TPR	AUC	TPR
P.D. 70M	Edgington	{81,100,9}	{17.5,25,9}	{58.4,0,0}	{4.1,100,18}	{54.2,25,9}	**{4,100,72}**
	Fisher	{24.4,25,72}	{0.2,25,72}	{42.9,100,72}	{0.8,25,72}	{47.7,100,72}	{1.8,25,36}
	Pearson	{79.7,0,0}	{16.2,25,9}	{57.9,25,9}	{3,25,9}	**{54.3,100,18}**	{3.5,25,9}
	George	**{81.5,0,0}**	**{28.9,25,9}**	**{58.6,0,0}**	**{5.3,0,0}**	{54.1,0,0}	{3.3,100,72}
	LR + PCA	{81,0,0}	{21.5,0,0}	{55.1,100,2}	{3.6,0,0}	{52.2,0,0}	{1.3,0,0}
P.D. 160M	Edgington	{81.2,0,0}	{21.8,25,9}	{58.6,0,0}	{3.6,0,0}	{54.8,0,0}	**{4.6,25,72}**
	Fisher	{24.3,25,72}	{0.2,25,72}	{43.5,100,72}	{0.8,25,72}	{4.7,25,72}	{1.7,100,72}
	Pearson	{79.2,0,0}	{15.4,0,0}	{58.1,0,0}	{2.5,0,0}	{54.7,100,9}	{3.9,100,9}
	George	**{81.8,0,0}**	**{29.5,25,18}**	**{58.9,0,0}**	**{5.9,0,0}**	**{54.9,0,0}**	{3.8,25,36}
	LR + PCA	{81.6,0,0}	{25.7,0,0}	{56.7,100,4}	{3.8,0,0}	{53.7,0,0}	{1.9,0,0}
P.D. 1.4B	Edgington	{79.3,0,0}	{12.3,0,0}	{59.7,0,0}	**{5.6,0,0}**	**{55.9,25,18}**	**{5.9,100,36}**
	Fisher	{27.6,100,72}	{0.2,100,18}	{43.2,25,72}	{0.8,25,9}	{44.3,0,0}	{1.4,100,72}
	Pearson	{77.8,0,0}	{10.6,0,0}	{58.9,0,0}	**{6.2,0,0}**	{55.7,25,36}	{4.2,100,36}
	George	{79.7,0,0}	**{21.9,0,0}**	**{59.9,0,0}**	{6.1,0,0}	**{55.9,25,36}**	{6.1,100,18}
	LR + PCA	**{80.8,100,1}**	{21.2,100,2}	{57.7,0,0}	{4.9,0,0}	{54.9,0,0}	{4.4,25,4}
G.N. 125M	Edgington	**{83,25,9}**	**{28,25,9}**	**{58.3,0,0}**	{3.8,0,0}	{54.3,0,0}	{3.6,25,72}
	Fisher	{21.1,100,72}	{0.4,25,72}	{43.3,25,72}	{0.7,0,0}	{46.6,25,72}	{1.4,0,0}
	Pearson	{81.5,25,9}	{26.5,25,9}	{57.9,0,0}	{3,0,0}	{53.7,100,9}	{2.5,25,72}
	George	{82.9,0,0}	**{31.6,25,18}**	{58.2,0,0}	{4,25,9}	**{54.5,100,9}**	**{4.4,100,36}**
	LR + PCA	{81.8,0,0}	{18.3,0,0}	{57,25,4}	{2.4,0,0}	{53.1,0,0}	{2,0,0}
G.N. 1.3B	Edgington	{81.3,100,9}	{19.9,0,0}	**{59.5,0,0}**	{4.7,0,0}	{55.1,100,18}	**{5.9,100,72}**
	Fisher	{22.4,25,72}	{0.6,25,72}	{43.2,100,72}	{1.4,25,72}	{45.3,100,72}	{0.9,100,72}
	Pearson	{79.9,0,0}	{21.1,100,9}	{58.9,25,9}	**{5.5,0,0}**	{54.8,25,18}	{4.8,25,72}
	George	{81.4,0,0}	**{25.4,0,0}**	{59.4,0,0}	{4,0,0}	**{55.2,25,36}**	{5.6,100,9}
	LR + PCA	**{81.5,100,1}**	{19.4,0,0}	{55.5,100,1}	{3.3,100,2}	{54.5,0,0}	{3.3,0,0}

Section 4.2 contains a compilation of the best-performing experiment results. It can be checked whether the configuration in experiment 5 was optimal compared to the configuration of the other experiments, and if so whether it improved upon the results in the original CAMIA paper (11) in an absolute sense. Aggregate results of hyperparameter configurations are available in Section 4.2.6, and analyses and discussions of the results are available in Chapter 5.

4.2 Best-performing experiments

4.2.1 Overall best

The following tables report the best-performing experiment results for each target model and dataset combination. They specify the strongest performing attack(s) and which experiment it corresponds to. It also shows the AUC-ROC and TPR@1%FPR scores with the best hyperparameter configuration. Each tuple contains the following values:

1. Attack type ∈ {E: Edgington, F: Fisher, P: Pearson, G: George, LR: Logistic Regression.} (described in Sections 2.2.2.2 and 2.2.2.3)
2. Experiment number ∈ $\{1, 2, 3, 4, 5\}$ (described in Section 4.1)
3. Experiment score ∈ $[0, 1]$
4. Whether the result is better than in CAMIA (11) in absolute terms ∈ {N,S,Y} (see tables A.1 and A.2)

Table 4.11: Best overall configurations on ArXiv and DM Mathematics

Model (size)	ArXiv		DM Mathematics	
	AUC	TPR	AUC	TPR
P.D. 70M	{G,3,76.4,N}	{G,4–5,21.3,N}	{G,1–5,95.0,S}	{G,4–5,77.5,N}
P.D. 160M	{G,3–5,78.5,N}	{LR,1,26.0,Y}	{G,1–2,95.0,S}	{LR,4,71.9,N}
P.D. 1.4B	{G,4–5,80.7,N}	{LR,5,32.1,Y}	{LR,4,94.1,N}	{LR,4,51.7,N}
G.N. 125M	{G,3,80.5,Y}	{E,1,29.7,Y}	{G,1,95.0,S}	{LR,2,60.8,N}
G.N. 1.3B	{G,3–5,81.6,N}	{LR,1–2,32.0,N}	{LR,3–5,93.5,N}	{LR,1–2,61.0,N}

Table 4.12: Best overall configurations on GitHub and PubMed Central

Model (size)	GitHub		PubMed-C.	
	AUC	TPR	AUC	TPR
P.D. 70M	{LR,1–2,85.9,N}	{E,2,51.6,N}	{LR/G,1–2/4–5,81.5,N}	{G,4–5,28.9,Y}
P.D. 160M	{E/G,1–5,86.5,N}	{E,5,56.6,N}	{LR,1–2,82.3,N}	{G,1,30.0,N}
P.D. 1.4B	{E/G,1–5,89.1,N}	{G,1,59.7,Y}	{LR,2/5,80.8,N}	{G/LR,1–2,22.0,N}
G.N. 125M	{LR,2,88.0,Y}	{LR,2,62.1,Y}	{E,1,83.1,Y}	{G,3,31.1,Y}
G.N. 1.3B	{G,1–5,89.6,N}	{G/LR,1–2,64.0,N}	{LR,1–2,82.1,N}	{G,1–2,26.0,N}

Table 4.13: Best overall configurations on HackerNews and Pile-CC

Model (size)	HackerNews		Pile-CC	
	AUC	TPR	AUC	TPR
P.D. 70M	{G,4–5,58.6,N}	{G,4–5,5.3,Y}	{P,1–3,54.4,Y}	{E,4,4.1,Y}
P.D. 160M	{G,4–5,58.9,N}	{G,1–2,6.0,Y}	{G,3–5,54.9,Y}	{E,1,4.8,Y}
P.D. 1.4B	{G,3–5,59.9,N}	{P,4–5,6.2,Y}	{E/G,1/2/4/5,55.9,Y}	{E,4,6.3,Y}
G.N. 125M	{E,2–3,58.2,N}	{E,4,4.1,N}	{G,1/3,54.5,Y}	{G,1/5,4.5,Y}
G.N. 1.3B	{E,5,59.5,N}	{P,3–5,5.5,Y}	{G,2/5,55.2,Y}	{G,1–2,6.0,N}

4.2.2 Edgington test comparison

Since CAMIA consists of two separate tests with different levels of data access, results for each setting should be assessed. The first, where the adversary only has access to non-member samples, is evaluated with the Edgington method exclusively in CAMIA (11) (See Section 2.2.2.2 for a description of the approach). The following tables report the best-performing Edgington experiment results for each target model and dataset combination. They specify which experiment it corresponds to, as well as the AUC-ROC and TPR@1%FPR scores with the best hyperparameter configuration, in each case. Each tuple contains the following values:

1. Experiment number $\in \{1, 2, 3, 4, 5\}$
2. Experiment score $\in [0, 1]$
3. Number of neighbours $n \in \{0, 25, 100\}$
4. Neighbourhood signal weight $w \in \{0, 9, 18, 36, 72\}$.
5. Whether the result is better than in CAMIA (11) in absolute terms $\in$ {N,S,Y} (see tables A.1 and A.2)

Table 4.14: Best Edgington configurations on ArXiv and DM Mathematics

Model (size)	ArXiv		DM Mathematics	
	AUC	TPR	AUC	TPR
P.D. 70M	{3,76.2,0,0,N}	{4–5,15.8,0,0,N}	{1–2,93.4,0,0,Y}	{4–5,54.4,0,0,N}
P.D. 160M	{3–5,78.3,0,0,N}	{3,21.5,0,0,N}	{1–2,90.9,0,0,Y}	{1–2,28.8,0,0,N}
P.D. 1.4B	{1–3,80.5,0,0,N}	{2,24.0,0,0,N}	{1–2,82.5,0,0,N}	{3–5,8.4,0,0,N}
G.N. 125M	{3,80.2,0,0,N}	{1,29.7,0,0,Y}	{1,91.3,0,0,Y}	{2,32.2,0,0,Y}
G.N. 1.3B	{3–5,81.5,0,0,Y}	{3–4,31.8,0,0,Y}	{1–2,86.7,0,0,Y}	{1–2,21.0,0,0,Y}

Table 4.15: Best Edgington configurations on GitHub and PubMed Central

Model (size)	GitHub		PubMed-C.	
	AUC	TPR	AUC	TPR
P.D. 70M	{1–2,85.4,100/25,18,Y}	{2,51.6,25,36,Y}	{1,81.1,100,9,Y}	{2,20.6,25,9,Y}
P.D. 160M	{1–5,86.5,25/100,9,N}	{5,56.6,25,18,Y}	{1–2,81.3,0,0,Y}	{1,23.0,100,9,Y}
P.D. 1.4B	{1–5,89.1,100/25,9,Y}	{2,59.0,100,9,Y}	{1–2,79.4,0,0,Y}	{1–2,14.0,0,0,N}
G.N. 125M	{2,87.0,25,9,S}	{1,52.4,25,18,Y}	{1,83.1,25,9,Y}	{1,28.8,25,9,Y}
G.N. 1.3B	{1–4,89.4,100/25,9,N}	{1–2,60.0,100,9,N}	{1,81.5,100,9,Y}	{1–2,23.0,0,0,Y}

Table 4.16: Best Edgington configurations on HackerNews and Pile-CC

Model (size)	HackerNews		Pile-CC	
	AUC	TPR	AUC	TPR
P.D. 70M	{4–5,58.4,0,0,N}	{1,4.6,25,9,N}	{1–3,54.3,0,0,Y}	{4,4.1,25,72,Y}
P.D. 160M	{3–5,58.6,0,0,N}	{1–5,3.6,0,0,Y}	{3–5,54.8,0,0,Y}	{1,4.8,100,72,Y}
P.D. 1.4B	{3–5,59.7,0,0,N}	{2,6.0,0,0,Y}	{2/4–5,55.9,100/25,18,Y}	{4,6.3,100,72,Y}
G.N. 125M	{5,58.3,0,0,N}	{2/4,41.0,0,0,Y}	{1,54.3,0,0,Y}	{4,3.8,100,72,N}
G.N. 1.3B	{5,59.5,0,0,N}	{3–5,4.7,0,0,Y}	{1–5,55.1,25/100,9/18,Y}	{5,5.9,100,72,N}

4.2.3 Logistic regression test comparison

Since CAMIA consists of two separate tests with different levels of data access, results for each setting should be assessed. The second, where the adversary has access to both member and non-member samples, is evaluated with logistic regression and group-based principal component analysis in CAMIA (11) (See Section 2.2.2.3 for a description of the approach). The following tables report the best-performing logistic regression experiment results for each target model and dataset combination. They specify which experiment it corresponds to, as well as the AUC-ROC and TPR@1%FPR scores with the best hyperparameter configuration, in each case. Each tuple contains the following values:

1. Experiment number $\in \{1, 2, 3, 4, 5\}$
2. Experiment score $\in [0, 1]$
3. Number of neighbours $n \in \{0, 25, 100\}$
4. Neighbourhood signal feature importance $\gamma \in \{0, 1, 2, 3, 4\}$.
5. Whether the result is better than in CAMIA (11) in absolute terms $\in$ {N,S,Y} (see tables A.1 and A.2)

Table 4.17: Best logistic regression configurations on ArXiv and DM Mathematics

Model (size)	ArXiv		DM Mathematics	
	AUC	TPR	AUC	TPR
P.D. 70M	{1–2,76.3,0,0,N}	{1–2,16.6,25,1/3,N}	{4,93.6,0,0,N}	{1–2,64.4,0,0,N}
P.D. 160M	{1,78.4,0,0,N}	{1,26.0,100,4,Y}	{4,94.6,100,1,N}	{4,71.9,100,2,N}
P.D. 1.4B	{1–2,80.2,0,0,N}	{5,32.1,25,3,Y}	{4,94.1,100,1,N}	{4,51.7,25,3,N}
G.N. 125M	{1,79.8,0,0,N}	{2,26.5,0,0,N}	{1,92.8,25,1,N}	{2,60.8,25,1,N}
G.N. 1.3B	{1–2,81.1,0,0,N}	{1–2,32.0,0,0,N}	{3–5,93.5,0,0,N}	{3–5,57.0,0,0,N}

Table 4.18: Best logistic regression configurations on GitHub and PubMed Central

Model (size)	GitHub		PubMed-C.	
	AUC	TPR	AUC	TPR
P.D. 70M	{1–2,85.9,0,0,N}	{4–5,49.5,25,1,N}	{1–2,81.5,0,0,N}	{1–2,25.6,0,0,N}
P.D. 160M	{1–2,86.4,0,0,N}	{3,46.7,100,4,N}	{1–2,82.3,100/25,1,N}	{1–2,28.9,0,0,N}
P.D. 1.4B	{3–5,88.5,0,0,N}	{1,34.4,0,0,N}	{2/5,80.8,100,3/1,N}	{2,22.0,25,2,N}
G.N. 125M	{2/4,88.0,0,0,N}	{2,62.1,25,1,N}	{2,82.7,0,0,N}	{2,24.8,100,1,N}
G.N. 1.3B	{1–2,89.3,0,0,N}	{1–2,64.0,0,0,N}	{1–2,82.1,0/100,0/1,N}	{1–2,21.0,0,0,N}

Table 4.19: Best logistic regression configurations on HackerNews and Pile-CC

Model	HackerNews		Pile-CC	
	AUC	TPR	AUC	TPR
P.D. 70M	{1,55.9,100,2,N}	{4–5,3.6,0,0,N}	{4–5,52.2,0,0,Y}	{4–5,1.3,0,0,Y}
P.D. 160M	{1/3,56.9,100,2,N}	{1–2,4.2,0,0,N}	{4–5,53.7,0,0,Y}	{1–2,2.0,25/0,3/0,Y}
P.D. 1.4B	{3–5,57.7,100/0,1/0,N}	{3–5,4.9,0,0,Y}	{1–2,55.2,0,0,Y}	{5,4.4,25,4,Y}
G.N. 125M	{1,57.8,100,2,Y}	{1–3,2.5,25/100,72,N}	{3,53.1,0,0,Y}	{1–5,2.0,0/25,0,Y}
G.N. 1.3B	{1,56.0,100,3,N}	{3–4,3.4,25,1,N}	{3–5,54.5,0,0,Y}	{3–4,3.4,0,0,Y}

4.2.4 Individual AUC-ROC and TPR@1%FPR results

The following table reports AUC-ROC and TPR@1%FPR results for each individual new signal, for each dataset and model experimented upon. This is important to distinguish which signals work well in isolation, and whether there are any meaningful differences between datasets and/or models. An AUC-ROC score equal to or less than 0.5, and a TPR@1%FPR score equal to or less than 1, corresponds to random or worse results, whereas higher scores indicate that the signal is useful for membership inference. Note that the signal $f_{nei\_standard}$ refers to the neighbourhood comparison signal in experiments 1 and 3 (see sections 4.1.1 and 4.1.3). Similarly, $f_{nei\_k\_highest}$ corresponds to experiment 2 (see section 4.1.2), f_{nei_elbow} to experiment 4 (see section 4.1.4) and f_{nei_flat} to experiment 5 (see section 4.1.5). A discussion on the results in Table 4.20 is available in Section 5.2.1.

Table 4.20: AUC-ROC and TPR@1%FPR results for each novel signal

Signal	ArXiv		Math		GitHub		PubMed-C.		HackerN.		Pile-CC	
	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
Pythia-D. 70M												
$f_{\text{nei\_standard}}$	0.609	3.314	0.516	2.698	0.796	0.106	0.569	0.959	0.528	1.611	0.512	1.371
$f_{\text{nei\_k\_highest}}$	0.611	3.486	0.513	1.746	0.795	0.106	0.565	0.959	0.531	1.656	0.514	1.386
$f_{\text{nei\_elbow}}$	0.609	3.571	0.515	1.270	0.794	0.000	0.563	1.047	0.532	1.567	0.513	1.314
$f_{\text{nei\_flat}}$	0.608	3.343	0.513	2.698	0.796	0.106	0.563	1.308	0.534	1.523	0.514	1.271
$f_{\text{elbow\_frac}}$	0.596	4.257	0.323	0.317	0.521	0.000	0.549	0.872	0.548	1.854	0.482	1.814
$f_{\text{elbow\_delta\_slope}}$	0.509	0.286	0.476	0.794	0.516	0.000	0.431	0.407	0.505	1.192	0.505	1.071
$f_{\text{flat\_frac}}$	0.575	6.114	0.408	0.000	0.461	0.053	0.551	1.337	0.502	1.249	0.527	1.557
Pythia-D. 160M												
$f_{\text{nei\_standard}}$	0.625	6.286	0.499	2.698	0.767	0.106	0.567	1.047	0.517	1.501	0.514	1.957
$f_{\text{nei\_k\_highest}}$	0.627	5.800	0.494	2.698	0.765	0.106	0.563	1.076	0.520	1.347	0.516	1.871
$f_{\text{nei\_elbow}}$	0.628	5.514	0.491	2.698	0.765	0.106	0.558	1.076	0.521	1.214	0.516	1.857
$f_{\text{nei\_flat}}$	0.626	5.629	0.490	2.698	0.766	0.106	0.559	1.076	0.522	1.413	0.516	2.000
$f_{\text{elbow\_frac}}$	0.619	3.786	0.317	0.000	0.526	0.000	0.537	0.581	0.549	2.209	0.489	1.129
$f_{\text{elbow\_delta\_slope}}$	0.514	2.286	0.544	3.175	0.555	5.426	0.554	1.395	0.501	2.274	0.496	0.671
$f_{\text{flat\_frac}}$	0.583	6.579	0.357	0.000	0.502	0.160	0.528	0.985	0.504	2.553	0.534	1.843
Pythia-D. 1.4B												
$f_{\text{nei\_standard}}$	0.658	7.743	0.534	3.651	0.716	0.000	0.549	0.930	0.506	1.854	0.542	2.600
$f_{\text{nei\_k\_highest}}$	0.661	9.257	0.530	3.651	0.713	0.000	0.545	0.872	0.510	1.766	0.545	2.643
$f_{\text{nei\_elbow}}$	0.659	9.029	0.531	2.698	0.713	0.000	0.545	0.872	0.512	1.634	0.545	2.786
$f_{\text{nei\_flat}}$	0.660	9.000	0.530	3.651	0.707	0.000	0.537	0.988	0.512	1.876	0.545	2.800
$f_{\text{elbow\_frac}}$	0.622	3.200	0.289	0.000	0.549	0.000	0.520	0.349	0.549	2.508	0.487	1.786
$f_{\text{elbow\_delta\_slope}}$	0.532	3.286	0.567	3.016	0.624	4.202	0.571	3.372	0.505	1.656	0.505	0.386
$f_{\text{flat\_frac}}$	0.595	6.529	0.305	0.000	0.534	0.053	0.557	1.463	0.509	2.121	0.504	1.300
GPT-Neo 125M												
$f_{\text{nei\_standard}}$	0.636	5.229	0.467	2.381	0.754	0.000	0.646	1.163	0.530	1.921	0.513	2.943
$f_{\text{nei\_k\_highest}}$	0.639	6.029	0.466	2.381	0.752	0.000	0.644	0.872	0.532	1.965	0.515	2.829
$f_{\text{nei\_elbow}}$	0.638	5.457	0.465	2.381	0.753	0.000	0.640	0.872	0.532	1.810	0.516	2.771
$f_{\text{nei\_flat}}$	0.638	5.543	0.465	2.381	0.749	0.000	0.643	0.959	0.534	2.252	0.515	2.571
$f_{\text{elbow\_frac}}$	0.663	5.500	0.313	0.000	0.540	0.160	0.573	0.669	0.556	1.495	0.484	1.957
$f_{\text{elbow\_delta\_slope}}$	0.550	1.514	0.547	3.175	0.664	9.415	0.509	1.541	0.506	1.567	0.516	1.314
$f_{\text{flat\_frac}}$	0.596	10.329	0.338	0.000	0.502	0.266	0.554	2.173	0.511	2.010	0.521	1.686
GPT-Neo 1.3B												
$f_{\text{nei\_standard}}$	0.661	5.343	0.460	0.794	0.714	0.000	0.641	0.843	0.515	1.987	0.533	3.557
$f_{\text{nei\_k\_highest}}$	0.665	5.486	0.458	0.794	0.712	0.000	0.637	0.901	0.519	2.208	0.535	3.586
$f_{\text{nei\_elbow}}$	0.663	5.457	0.453	0.794	0.713	0.000	0.633	0.872	0.523	2.053	0.534	3.900
$f_{\text{nei\_flat}}$	0.663	4.743	0.461	0.635	0.707	0.000	0.633	0.930	0.520	2.274	0.535	3.686
$f_{\text{elbow\_frac}}$	0.649	4.086	0.282	0.000	0.575	0.419	0.585	0.552	0.543	1.271	0.493	2.086
$f_{\text{elbow\_delta\_slope}}$	0.577	2.771	0.595	1.270	0.737	6.755	0.548	2.878	0.515	2.517	0.521	1.129
$f_{\text{flat\_frac}}$	0.596	7.129	0.288	0.000	0.579	0.053	0.553	0.707	0.495	2.497	0.507	2.114

4.2.5 Average slope profile (member vs. non-member)

There are claims in (11) that the decreasing rates for the loss sequence of members and non-members are different, and that the rate of a member is significantly higher than a non-member. To confirm whether these claims are true, analyses across datasets and models were performed to measure the average slope of the ordinary least squares (OLS) line fitted to the next-token loss sequences of each member and non-member of each dataset and model. As can be seen in Table 4.21 below, the data do not support the claims. Further analysis of these results is available in Section 5.2.2.

Table 4.21: Mean OLS slope of next-token loss sequences

Model	ArXiv		Math		GitHub		PubMed-C.		HackerN.		Pile-CC	
	mem	non	mem	non	mem	non	mem	non	mem	non	mem	non
P.D. 70M	-0.0025	-0.0035	-0.0014	-0.0011	-0.0006	-0.0037	-0.0003	-0.0020	-0.0015	-0.0018	-0.0031	-0.0033
P.D. 160M	-0.0023	-0.0031	-0.0012	-0.0009	-0.0007	-0.0035	-0.0003	-0.0019	-0.0011	-0.0015	-0.0030	-0.0030
P.D. 1.4B	-0.0021	-0.0030	-0.0009	-0.0007	-0.0004	-0.0033	-0.0001	-0.0016	-0.0006	-0.0010	-0.0026	-0.0027
G.N. 125M	-0.0015	-0.0024	-0.0010	-0.0008	-0.0004	-0.0030	0.0001	-0.0015	-0.0008	-0.0011	-0.0029	-0.0030
G.N. 1.3B	-0.0014	-0.0024	-0.0008	-0.0007	-0.0003	-0.0027	0.0001	-0.0013	-0.0002	-0.0005	-0.0026	-0.0027

An example of the distribution and average of whole-sequence slopes for members and non-members of a certain dataset on a particular model is illustrated in Figure 4.1. The OLS slopes of the GitHub dataset for GPT-Neo 1.3B are steeper for non-members than members, contradicting the notion that member slopes are steeper than non-member slopes.

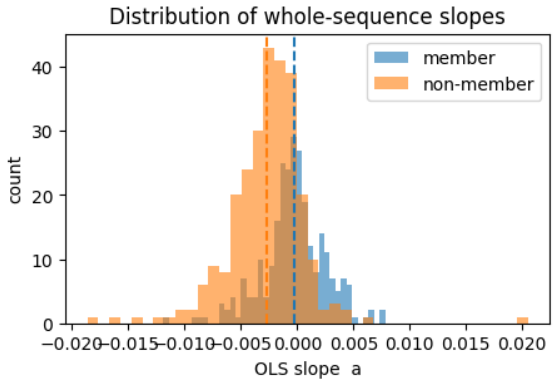


Figure 4.1: OLS slopes of the GitHub dataset for GPT-Neo 1.3B

4.2.6 Hyperparameter results

The following table shows what proportion of experiments where the neighbourhood signal was included produced the best results, for each attack type and model. The results are aggregated across datasets and metrics (AUC-ROC and TPR@1%FPR are counted simultaneously).

Table 4.22: Neighbourhood signal inclusion optimality proportions

Exp.	Attack	P.D. 70M	P.D. 160M	P.D. 1.4B	G.N. 125M	G.N. 1.3B
1	Edgington	50.0	33.3	33.3	50.0	41.7
	Fisher	91.7	91.7	91.7	75.0	75.0
	Pearson	58.3	33.3	41.7	58.3	75.0
	George	41.7	33.3	41.7	50.0	41.7
	Logistic Regression	25.0	58.3	58.3	50.0	16.7
2	Edgington	50.0	25.0	33.3	41.7	41.7
	Fisher	91.7	91.7	83.3	58.3	83.3
	Pearson	58.3	33.3	50.0	58.3	66.7
	George	41.7	25.0	41.7	50.0	33.3
	Logistic Regression	33.3	50.0	58.3	50.0	25.0
3	Edgington	50.0	33.3	33.3	41.7	50.0
	Fisher	91.7	91.7	75.0	58.3	75.0
	Pearson	58.3	33.3	41.7	66.7	66.7
	George	41.7	25.0	41.7	41.7	50.0
	Logistic Regression	8.3	33.3	41.7	33.3	16.7
4	Edgington	50.0	33.3	33.3	41.7	41.7
	Fisher	91.7	91.7	83.3	66.7	75.0
	Pearson	50.0	50.0	50.0	58.3	66.7
	George	25.0	25.0	33.3	41.7	25.0
	Logistic Regression	16.7	41.7	41.7	41.7	33.3
5	Edgington	58.3	33.3	41.7	41.7	41.7
	Fisher	91.7	91.7	83.3	58.3	75.0
	Pearson	58.3	41.7	50.0	58.3	58.3
	George	25.0	25.0	33.3	41.7	33.3
	Logistic Regression	16.7	33.3	50.0	25.0	33.3
All	Edgington	51.7	31.7	35.0	43.3	43.3
	Fisher	91.7	91.7	83.3	63.3	76.7
	Pearson	56.7	41.7	46.7	60.0	66.7
	George	35.0	25.0	38.3	45.0	36.7
	Logistic Regression	18.3	43.3	50.0	40.0	25.0

The following table shows the frequency of each neighbourhood-signal weight W (for the p-value tests) or feature–importance γ (for the LR + PCA attack) being optimal in each experiment, given that they are not 0.

Table 4.23: Optimal weights and feature importances

Experiment	Weight W				Feature–importance γ			
	$W = 9$	$W = 18$	$W = 36$	$W = 72$	$\gamma = 1$	$\gamma = 2$	$\gamma = 3$	$\gamma = 4$
1	38.64	17.42	6.82	37.12	48.00	28.00	12.00	12.00
2	38.58	13.39	10.24	37.79	56.00	8.00	32.00	4.00
3	37.01	16.54	8.66	37.79	35.29	23.53	23.53	17.65
4	33.87	15.32	12.10	38.71	47.61	19.05	19.05	14.29
5	35.49	15.32	10.48	38.71	42.11	31.58	5.26	21.05
Total	36.75	15.62	9.62	38.01	46.73	21.50	18.69	13.08

The following table shows the proportions (in percent) of the best performing attack types for each model and experiment.

Table 4.24: Best attacks in each experiment

Exp.	Attack	P.D. 70M	P.D. 160M	P.D. 1.4B	G.N. 125M	G.N. 1.3B
1	Edgington	15.4	21.4	0	33.3	14.3
	Fisher	0	0	0	0	0
	Pearson	7.7	0	0	0	7.1
	George	53.8	50.0	66.7	58.3	57.1
	Logistic Regression	23.1	28.6	33.3	8.3	21.4
2	Edgington	7.7	21.4	26.3	23.1	6.3
	Fisher	0	0	0	0	0
	Pearson	15.4	0	5.3	0	12.5
	George	53.8	50.0	42.1	53.8	0.5
	Logistic Regression	23.1	28.6	26.3	23.1	31.3
3	Edgington	16.7	23.1	14.3	28.6	25.0
	Fisher	0	0	0	0	0
	Pearson	8.3	0	0	0	6.3
	George	59.7	61.5	57.1	64.3	56.3
	Logistic Regression	14.5	15.4	28.6	7.1	12.5
4	Edgington	16.7	16.7	26.7	30.8	21.4
	Fisher	0	0	0	0	0
	Pearson	8.3	0	6.7	0	7.1
	George	66.7	66.7	40.0	53.8	57.1
	Logistic Regression	8.3	16.7	26.7	15.4	14.3
5	Edgington	16.7	16.7	26.7	41.7	25.0
	Fisher	0	0	0	0	0
	Pearson	16.7	0	6.7	0	8.3
	George	66.7	66.7	40.0	58.3	50.0
	Logistic Regression	0	16.7	26.7	0	16.7
All	Edgington	14.5	20.0	20.0	31.3	18.1
	Fisher	0	0	0	0	0
	Pearson	11.3	0	4.0	0	8.3
	George	66.7	58.5	48.0	57.8	54.2
	Logistic Regression	8.3	21.5	28.0	10.9	19.4

5

Discussion and Conclusion

5.1 Experimental results

The results from the five experiments are mixed. Some attacks do not perform better than the original CAMIA (11) implementation on some datasets and models, other outperform it and could potentially be considered state-of-the-art membership inference attacks. An important consideration is that many of the stronger results only consist of marginally higher scores, while a few are significantly higher. Interestingly, each of the five attacks outperform both CAMIA and the other four novel attacks on at least one dataset/model combination. This implies that adding new signals and/or new types of semantically perturbed neighbours do not always generate stronger results, although this is occasionally the case.

There are clear differences in performance depending on dataset, which is to be expected due to the small sample sizes and their relative differences, as well as the inherently different structure and syntax of the language in the various sets (19). Results mostly improve with increased model size, in accordance with standard results in the field. This warrants further analysis on larger models, particularly the ones experimented on in (11) which were excluded from this study due to time constraints.

It is crucial that the implementation of CAMIA (11) is custom and not based on the authors' original code. The technical details concerning, for instance, logistic regression and principal component analysis hyperparameters are not documented and the differences in implementation could affect overall results to an unknown extent. A distinction must therefore be made between whether the attacks in the five experiments attain absolutely higher scores than CAMIA, and whether the addition of novel signals or neighbours improve upon the baseline custom implementation. If the latter is true, it indicates that the inclusion of such signals and/or neighbours could improve results in general, for different types of implementations.

The characteristics of the overall best results (Section 4.2.1) differ between datasets. The new attacks performed better than CAMIA both in terms of AUC-ROC and TPR@1%FPR on the Pile-CC dataset with essentially all models. This is promising, since it has the largest sample size out of all used subsets. Results were worse for other datasets, but scores higher than CAMIA were achieved on all datasets and metrics for at least one model, except for AUC-ROC on the HackerNews dataset,

and both metrics for the DM Mathematics dataset. Results were generally good for the Edgington attack compared to CAMIA, particularly for the PubMed Central and Pile-CC datasets, where the majority of scores were higher on all models. The logistic regression attack performed significantly worse compared to CAMIA, with most scores being lower than in the original paper. An exception was again the Pile-CC dataset, where the novel attacks was higher for both metrics on all models.

A concrete example of a substantial improvement is that on GitHub with Pythia-deduped 160M in the CAMIA Edgington attack, TPR@1%FPR increased from 41.8% to 56.6%. Another was with the Pythia-deduped 1.4B model, TPR@1%FPR increased from 2.74% to 4.4% (a 60% increase).

5.2 Analysis of additional results

5.2.1 Individual signals performances

Table 4.20 shows AUC-ROC and TPR@1%FPR of each new signal on each dataset and model. The signals perform differently on the various datasets. Both AUC and TPR scores on ArXiv are consistently high, with the neighbourhood comparison signal with flatline neighbours f_{flat_frac} (see section 3.4.3) achieving 10.329 TPR on the GPT-Neo 125M model, and the neighbourhood comparison signal with the K highest-loss neighbours $f_{nei\_k\_highest}$ (see section 3.4) 0.665 on the GPT-Neo 1.3B model. Performance on the DM Mathematics dataset is mostly poor, which can probably be attributed to the very small sample size. Interestingly, AUC scores for GitHub are consistently high, whereas TPR scores are very low and often 0. However, the absolute magnitude of maximal slope difference signal $f_{elbow\_delta\_slope}$ (Section 3.4.2) is an exception, mostly achieving high scores (e.g. 9.415 on the GPT-Neo 125M model). AUC scores for the PubMed Central dataset are better than random, with scores generally being in the 0.55-0.60 range. However, TPR scores are consistently performing as bad as random, again with the occasional exception of the $f_{elbow\_delta\_slope}$ signal. The relationship is mostly the opposite for the HackerNews dataset, where AUC scores are generally close to 0.5, whereas TPR scores are around 2. For the most challenging dataset, Pile-CC, AUC scores are mostly only slightly above 0.5, whereas TPR scores are around 2-3.

On an aggregate level, all signals display discriminatory power, indicating they could feasibly be implemented in future membership inference attacks. Varying results depending on dataset is in accordance with previous research in the field.

5.2.2 Average OLS slope of next-token loss sequences

Considering that the inspiration for the novel elbow and flatline signals is based on the claim by the authors of (11) that the magnitude of the rate of change for members on average is larger than for non-members, the claim should be verified. As can be seen in 4.21, the exact opposite is true in all cases. The exception is the DM Mathematics dataset. However, its strictly limited sample size of 89, limits significance. Although the intuition behind the original slope signals in CAMIA

and the novel signals in this thesis is incorrect, it should be noted that the signals are still evidently useful, as the data shows that they often can distinguish between members and non-members (see e.g. Table 4.20). This implies that as long as there on average is a difference between the member and non-member datasets, it can be leveraged as a membership inference attack signal. It also implies it is not strictly necessary to understand why this difference exists, as long as it can be identified (although a solid theoretical foundation of course is more robust). This could potentially influence the design of future membership inference attacks, where searches for arbitrary dataset-level differences could be conducted to build a bundle of membership distinguishing signals. This would lower the difficulty of designing powerful attacks, as they could essentially be randomized rather than theorized.

5.2.3 Hyperparameter results analysis

Tables 4.22, 4.23 and 4.24 show various hyperparameter related results. The first shows what proportion of experiments where the neighbourhood signal was included produced the best results, for each attack type and model. The results are also aggregated across datasets and metrics. An observation is that including the neighbourhood signal (thus implementing the Neighbourhood Comparison Attack in CAMIA) does not automatically improve performance. In many cases, including it deteriorates performance. The Fisher attack clearly benefits the most from its inclusion; this attack was however significantly weaker than all others in all experiments and never generated the highest score. Not including the neighbourhood signal in experiments 1 and 2 corresponds to the baseline (custom) CAMIA implementation. The data thus shows that the new attacks developed for this thesis do not always improve upon previous research. An important practical consideration is whether the neighbourhood signal should be implemented in an attack suite utilizing signals such as in the present case. Intuitively, if the signal improves results more often than not, it should be utilized. In all experiments, most attacks benefit from its inclusion less than 50% of the time. This implies there is limited support for its utilization in its current state. However, as all attacks in all experiments generated optimal results with the inclusion of the neighbourhood signal in a significant portion of cases, there is also reason to investigate the approach further. This could improve performance to an extent such that a neighbourhood signal should be incorporated in a state-of-the-art attack.

Concerning optimal weights and feature importances for the neighbourhood signal, table 4.23 implies that a smaller feature importance γ in the principal component analysis prior to applying logistic regression is mostly optimal. The neighbourhood signal weights in the classical statistical tests exclusively utilizing non-members generate the largest proportion of optimal results when it is set to either 9 or 72. These are the smallest and largest weights used in the experiments, respectively. There are 72 other signals in the test; setting the weight to this number is intended to correspond to equating the influence of the Neighbourhood Comparison Attack to CAMIA. Lower weights diminish this influence, whereas intermediate weights are rarely optimal.

A very interesting result is that George's method was the best attack in all experiments, by far (see table 4.24). This was one of the aggregation methods originally applied in CAMIA, although only Edgington's method was ultimately chosen for non-member experiments (11). The results in this thesis strongly indicate that switching to George's aggregation method could generally improve attack results. It should also be noted that Edgington's method does clearly outperform Fisher's and Pearson's methods. An important difference between the results from the experiments in this thesis and in CAMIA is that the proportion of instances where the logistic regression attack outperformed Edgington's method is significantly larger in the original CAMIA paper. This could possibly be explained by implementation differences, and that hyperparameter tuning could boost logistic regression performance for the novel attacks developed for this thesis.

In contrast to the findings of (16), the largest value (100) of the amount of neighbours used for the neighbourhood comparison signal was not always optimal, whenever using the signal generated better results than without. It is unclear why this difference exists. One theory is that the quality of neighbours deteriorates with an increased number due to a decreasing p_{swap} score, such that their influence becomes detrimental after a certain cut-off point.

5.3 Answers to research questions

The implementation of CAMIA with the NCA extension and novel elbow and flatline signals enabled five experiments testing the influence of targeted and loss-guided semantics-preserving token perturbations for MIAs. The attack architecture used also constitutes a union of recent MIAs, so the experiment results show whether improvements on previous results in the literature have been made. The research questions can therefore be answered below.

RQ1: For semantics-preserving text edits in membership inference attacks, can targeted and loss-guided token perturbations improve performance over masking random or arbitrary words?

The approach towards answering this question was by implementing the Neighbourhood Comparison Attack within the CAMIA architecture, and performing experiments with both standard neighbour generation using random masking (see algorithm 1), and two novel algorithms utilizing loss-guided and targeted perturbations (see algorithms 2 and 3). Experiments 1 and 2 (see sections 4.1.1 and 4.1.2) corresponds to baseline CAMIA extended with the Neighbourhood Comparison Attack, using standard neighbours in the former, and K highest loss token positions in the latter. Experiments 3, 4 and 5 (see sections 4.1.3, 4.1.4 and 4.1.5) correspond to baseline CAMIA extended with the Neighbourhood Comparison Attack as well as the elbow and flatline signals (see sections 3.4.2 and 3.4.3). Experiment 3 used standard neighbours, whereas the last two used elbow and flatline-based neighbours respectively. There is no clear answer to this research question based on the results, and conducting the experiments in this thesis is just one way of approaching the problem. It is clear that loss-guided, targeted perturbations performed better than

random masking in some cases, as experiments 2, 4 and 5 had the most optimal results in multiple settings, also sometimes with higher scores than in (11) (see tables 4.14, 4.15 and 4.16). On the other hand, there were also multiple cases where the results from experiments 1 and 3 had the most optimal results such that the randomly masked neighbours outperformed the targeted variants. In other words, there is some evidence that loss-guided, targeted neighbours outperform random masking for membership inference, but the question requires further examination for a more definitive answer.

An additional benefit of utilizing targeted perturbations such as in algorithms 2 and 3 is that the neighbour generation procedure can be executed significantly faster than algorithm 1. The original neighbour generation algorithm loops over every token of a text, whereas the new ones only operate on a limited set of tokens of a text, essentially reducing algorithmic complexity from $\mathcal{O}(n)$ to $\mathcal{O}(1)$. Merely achieving the same results with a more efficient algorithm could be an attractive prospect in practice.

RQ2: Can a unified attack that combines and develops recent MIAs outperform current state-of-the-art results?

In this thesis, an attack unifying the Context-aware Membership Inference Attack (CAMIA) (11) and the Neighbourhood Comparison Attack (NCA) (16) was developed. Furthermore, multiple new signals and neighbour generating algorithms were designed and incorporated (see section 3.4). CAMIA can be considered a state-of-the-art membership inference attack, given that it mostly outperforms previous MIAs. The results from the five experiments conducted in this thesis show that the new, unified attack often significantly outperform the results in CAMIA. The short answer to the research question is therefore that a unified attack which combines and develops recent MIAs can indeed outperform current state-of-the-art results. However, many improvements were marginal, and the new attack did not always outperform the baseline CAMIA implementation.

5.4 Limitations

Although the semantics-preserving Neighbourhood Comparison Attack was implemented, and two new algorithms generating semantic neighbours were designed for the novel membership inference attack, a limitation of this thesis was the otherwise limited focus on exploration of semantics as such. There is still wide room to explore the role and definition of semantics within the context of MIAs. A practical limitation also comes when the number of tokens to be perturbed to semantically equivalent ones is restricted. If the Gaussian in Algorithm 3 is narrow, the token at each valid index might need to be perturbed to a larger amount of variants than available synonyms, essentially invalidating the approach. As there is some evidence (16) for a larger amount of neighbours leading to stronger results, there might exist an optimal number of neighbours such that these factors become balanced. However, a smaller number of neighbours were often optimal in the experiments conducted in this thesis.

Another central limitation is that not all considered MIAs were incorporated in the

attack infrastructure. RaMIA and SMIA posits promising prospects for integration, but were due to time constraints not actually implemented. As combining MIAs has been shown to be effective in some cases, further integration of additional MIAs could be a feasible next step for this type of attack development.

Considering the large amount of experiments, datasets, models and hyperparameters combined with limited time and computational resources, not all variations could be experimented upon. Larger models needed to be excluded, as well as 13-gram variants of the datasets, various number of perturbed tokens per text, and so on.

Finally, larger datasets should be used in future experiments, as the limited sample sizes used for the experiments in this thesis could inflate variance - particularly for the TPR@1%FPR metric.

5.5 Practical Implications

For auditors who can query next-token losses but cannot train heavy reference models, augmenting CAMIA with the neighbourhood comparison signal is a low-cost win: it pushes TPR upward at fixed 1% FPR with negligible engineering overhead. The elbow and flatline signals are trivial to compute from a single pass of the loss sequence and can be layered in without affecting latency. Together, these yield a stronger, still lightweight pipeline that is appropriate for routine privacy risk assessments on pretrained LLMs.

5.6 Future work

There is clear potential for additional improvements of semantic membership inference attacks. A feasible starting point includes working on the limitations addressed above, for example by substituting multiple tokens simultaneously in the new attack. A concrete next step is to incorporate it within the LeakPro project infrastructure at AI Sweden. Development can be continued with increased resources, in order to experiment on larger models and implement RaMIA and SMIA, or any other feasible state-of-the-art MIAs. Another general direction could be to enable automatic identification of suitable substitution indices through machine learning rather than hand-crafted heuristics.

5.7 Conclusion

This thesis re-implemented CAMIA from its public description and showed that small, semantics-aware extensions can improve membership inference at the low-FPR operating points that matter in practice. This was performed by implementing a lightweight, reference-free neighbourhood comparison signal together with two simple next-token loss–trajectory descriptors (the elbow and flatline heuristics) and loss-guided neighbour generation. Across multiple model–dataset pairs (Pythia-deduped and GPT-Neo on six Pile subsets), the added signals consistently increased TPR

at fixed 1% FPR while leaving AUC-ROC largely unchanged, indicating better tail performance without heavy compute or supervision. These gains come from exploiting structure already present in next-token loss sequences and from probing the model with semantically near-equivalents rather than relying on verbatim matches. While the work focuses on local semantic variations and does not include formal significance tests, the results are reproducible, open-source, and practically useful for risk assessment under black-box loss access. Overall, the study demonstrates that pragmatic, low-cost semantics can make state-of-the-art MIAs more effective where a single false positive is expensive, and it opens clear paths toward broader semantic ranges, stronger neighbour generation, and integration with range-based MIAs.

Bibliography

[1] SoK: Sayanton V Dibbo. Model inversion attack landscape: Taxonomy, challenges, and future roadmap. In 2023 IEEE 36th Computer Security Foundations Symposium (CSF), pages 439–456. IEEE, 2023.

[2] AI Sweden et al. LeakPro: Leakage profiling and risk oversight of machine learning models. LeakPro, 2024. Accessed: 2024-09-17.

[3] W. Fu, et al. Membership Inference Attacks against Fine-tuned Large Language Models via Self-prompt Calibration. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[4] Bao-Ngoc Nguyen, Keshigeyan Chandrasegaran, Milad Abdollahzadeh, and Ngai-Man Man Che-ung. Label-only model inversion attacks via knowledge transfer. Advances in Neural Information Processing Systems, 36, 2024.

[5] G. Amit, A. Goldsteen, and A. Farkash. SoK: Reducing the Vulnerability of Fine-tuned Language Models to Membership Inference Attacks. *arXiv preprint arXiv:2403.08481*, 2024.

[6] M. Kaneko, et al. Sampling-based Pseudo-Likelihood for Membership Inference Attacks. *Findings of the Association for Computational Linguistics: ACL 2025, pages 8894–8907*, 2025.

[7] Xiaoxiao Sun, Nidham Gazagnadou, Vivek Sharma, Lingjuan Lyu, Hongdong Li, and Liang Zheng. Privacy assessment on reconstructed images: are existing evaluation metrics faithful to human perception? Advances in Neural Information Processing Systems, 36, 2024.

[8] NYT v. OpenAI: The Times's About-Face. *Harvard Law Review Blog*, 2024. `https://harvardlawreview.org/blog/2024/04/nyt-v-openai-the-timess-about-face/`.

[9] Du, Xiaoning and Xie, Xiaofei and Li, Yi and Ma, Lei and Liu, Yang and Zhao, Jianjun, DeepStellar: model-based quantitative analysis of stateful deep learning systems, Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (2019)

[10] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer. Membership

Inference Attacks From First Principles. *IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2022, pp. 1897-1914*, 2022.

[11] Chang, Hongyan, et al. "Context-Aware Membership Inference Attacks against Pre-trained Large Language Models." arXiv preprint arXiv:2409.13745 (2024).

[12] J. Tao and R. Shokri. Range Membership Inference Attacks. *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML), Copenhagen, Denmark, pp. 346-361*, 2025.

[13] H. Mozaffari and V. J. Marathe. Semantic Membership Inference Attack against Large Language Models. *arXiv:2406.10218v1 [cs.LG]*, 14 Jun 2024.

[14] Jie Zhang, Debeshee Das, Gautam Kamath, Florian Tramèr. Membership Inference Attacks Cannot Prove that a Model Was Trained On Your Data. *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML), Copenhagen, Denmark, pp. 333-345*, 2025.

[15] Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, Hannaneh Hajishirzi. Do Membership Inference Attacks Work on Large Language Models? *First Conference on Language Modeling*, 2024.

[16] J. Mattern, F. Mireshghallah, Z. Jin, B. Schölkopf, M. Sachan and T. Berg-Kirkpatrick. Membership Inference Attacks Against Language Models via Neighbourhood Comparison. *Findings of the Association for Computational Linguistics: ACL 2023*, 2023.

[17] justusmattern27, GitHub. `https://github.com/justusmattern27/neighbour-mia`, 2023.

[18] Hugging Face (2023) `https://huggingface.co/datasets/EleutherAI/pile/discussions/15`

[19] Gao, Leo and Biderman, Stella and Black, Sid and Golding, Laurence and Hoppe, Travis and Foster, Charles and Phang, Jason and He, Horace and Thite, Anish and Nabeshima, Noa and Presser, Shawn and Leahy, Connor. The Pile: An 800GB Dataset of Diverse Text for Language Modeling In *arXiv preprint arXiv:2101.00027*, 2020.

[20] Ji, Ziwei and Lee, Nayeon and Frieske, Rita and Yu, Tiezheng and Su, Dan and Xu, Yan and Ishii, Etsuko and Bang, Ye Jin and Madotto, Andrea and Fung, Pascale. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv. 55, 12, Article 248*, 2023.

[21] Yi Dong and Ronghui Mu and Gaojie Jin and Yi Qi and Jinwei Hu and Xingyu Zhao and Jie Meng and Wenjie Ruan and Xiaowei Huang. Position: Building Guardrails for Large Language Models Requires Systematic Design. *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[22] Wayne Xin Zhao and Kun Zhou and Junyi Li and Tianyi Tang and Xiaolei Wang and Yupeng Hou and Yingqian Min and Beichen Zhang and Junjie Zhang

and Zican Dong and Yifan Du and Chen Yang and Yushuo Chen and Zhipeng Chen and Jinhao Jiang and Ruiyang Ren and Yifan Li and Xinyu Tang and Zikang Liu and Peiyu Liu and Jian-Yun Nie and Ji-Rong Wen. A Survey of Large Language Models. *arXiv:2303.18223 [cs.CL]*, 2025.

[23] John X. Morris and Chawin Sitawarin and Chuan Guo and Narine Kokhlikyan and G. Edward Suh and Alexander M. Rush and Kamalika Chaudhuri and Saeed Mahloujifar. How much do language models memorize? *arXiv:2505.24832 [cs.CL]*, 2025.

[24] Mehdi Ali and Michael Fromm and Klaudia Thellmann and Richard Rutmann and Max Lübbering and Johannes Leveling and Katrin Klug and Jan Ebert and Niclas Doll and Jasper Schulze Buschhoff and Charvi Jain and Alexander Arno Weber and Lena Jurkschat and Hammam Abdelwahab and Chelsea John and Pedro Ortiz Suarez and Malte Ostendorff and Samuel Weinbach and Rafet Sifa and Stefan Kesselheim and Nicolas Flores-Herr. Tokenizer Choice For LLM Training: Negligible or Crucial? *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024.

[25] Andreas Lindholm, Niklas Wahlström, Fredrik Lindsten, Thomas B. Schön. MACHINE LEARNING - A First Course for Engineers and Scientists. *Cambridge University Press*, 2022.

[26] Weijia Shi and Anirudh Ajith and Mengzhou Xia and Yangsibo Huang and Daogao Liu and Terra Blevins and Danqi Chen and Luke Zettlemoyer. Detecting Pretraining Data from Large Language Models. *The Twelfth International Conference on Learning Representations*, 2024.

[27] Samuel Yeom and Irene Giacomelli and Matt Fredrikson and Somesh Jha. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. *2018 IEEE 31st Computer Security Foundations Symposium (CSF), Oxford, United Kingdom, pp. 268-282*, 2018.

[28] Nicholas Carlini and Florian Tramer and Eric Wallace and Matthew Jagielski and Ariel Herbert-Voss and Katherine Lee and Adam Roberts and Tom Brown and Dawn Song and Ulfar Erlingsson and Alina Oprea and Colin Raffel. Extracting Training Data from Large Language Models. *30th USENIX Security Symposium*, 2021.

[29] Jingyang Zhang and Jingwei Sun and Eric Yeats and Yang Ouyang and Martin Kuo and Jianyi Zhang and Hao Frank Yang and Hai Li. Min-K%++: Improved Baseline for Detecting Pre-Training Data from Large Language Models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[30] W. Zhou, T. Ge, K. Xu, F. Wei and M. Zhou. BERT-based Lexical Substitution. In *Proceedings of ACL 2019*, 2019.

A

Appendix

Table A.1: Original CAMIA results for the Pythia-Deduped suite (11)

Size	Attack	ArXiv		Mathematics		GitHub		PubMed		HackerNews		Pile-CC	
		AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
70M	Blind	0.73	0.00	0.86	0.00	0.82	0.00	0.71	0.00	0.52	0.88	0.53	0.56
	LOSS	0.73	6.97	0.95	78.73	0.82	19.52	0.79	15.47	0.58	3.05	0.53	1.94
	Zlib	0.73	7.23	0.82	28.41	0.86	48.94	0.78	14.01	0.57	2.72	0.51	3.23
	Min-K%	0.68	12.23	0.94	75.56	0.81	19.31	0.77	14.04	0.56	3.55	0.53	1.89
	Min-K%++	0.56	5.03	0.74	6.83	0.72	7.66	0.63	4.65	0.55	1.24	0.52	1.74
	Reference	0.52	2.80	0.63	2.06	0.68	4.31	0.66	6.16	0.52	0.46	0.50	2.13
	CAMIA (Edg.)	0.77	19.54	0.93	69.68	0.85	43.03	0.81	16.57	**0.59**	**4.83**	**0.53**	**4.01**
	CAMIA (LR/PCA)	**0.79**	**23.37**	**0.95**	**80.95**	**0.87**	**53.46**	**0.83**	**27.85**	**0.59**	3.64	0.51	1.09
160M	LOSS	0.74	7.26	0.94	70.32	0.83	24.31	0.79	18.28	0.57	2.91	**0.54**	2.70
	Zlib	0.74	6.06	0.81	21.59	0.87	45.48	0.78	16.74	0.57	2.76	0.52	3.37
	Min-K%	0.69	8.49	0.92	68.89	0.82	25.74	0.78	17.67	0.55	2.67	0.53	2.99
	Min-K%++	0.53	2.14	0.76	16.51	0.72	10.48	0.62	8.02	0.53	1.10	0.52	2.30
	Reference	0.57	1.09	0.62	0.16	0.68	3.51	0.68	4.04	0.51	1.04	0.52	2.64
	CAMIA (Edg.)	0.79	23.37	0.90	31.11	0.87	41.81	0.81	21.25	0.59	2.78	**0.54**	**4.67**
	CAMIA (LR/PCA)	**0.80**	**24.74**	**0.95**	**73.97**	**0.88**	**56.91**	**0.83**	**30.93**	**0.59**	**4.26**	0.53	1.40
1.4B	LOSS	0.77	12.63	0.92	43.49	0.86	30.05	0.78	16.16	0.59	1.99	0.55	4.56
	Zlib	0.77	9.83	0.80	15.24	0.89	36.38	0.77	13.95	0.58	2.19	0.54	5.91
	Min-K%	0.74	14.86	0.93	67.14	0.85	29.95	0.78	18.05	0.57	2.14	0.55	4.70
	Min-K%++	0.64	3.49	0.75	15.87	0.81	22.55	0.63	8.08	0.55	1.52	0.55	3.96
	Reference	0.71	6.66	0.50	1.27	0.72	0.96	0.67	1.54	0.54	1.39	**0.59**	5.90
	CAMIA (Edg.)	**0.81**	25.23	0.83	11.90	0.89	54.04	0.79	14.22	0.60	**4.99**	0.55	**6.03**
	CAMIA (LR/PCA)	**0.81**	**31.23**	**0.95**	**71.90**	**0.91**	**57.77**	**0.82**	**26.22**	**0.60**	4.55	0.55	2.74
2.8B	LOSS	0.78	14.11	0.91	19.21	0.87	39.68	0.78	18.28	0.60	2.03	0.55	4.63
	Zlib	0.77	10.86	0.80	11.43	0.90	42.02	0.77	14.51	0.59	2.49	0.54	5.83
	Min-K%	0.75	20.63	0.92	54.60	0.87	40.27	0.78	20.09	0.58	1.24	0.55	4.71
	Min-K%++	0.65	5.71	0.72	19.84	0.84	31.49	0.66	9.80	0.57	1.52	0.54	3.27
	Reference	0.71	6.46	0.45	0.79	0.72	4.79	0.63	1.60	0.57	3.00	**0.59**	6.34
	CAMIA (Edg.)	**0.81**	25.89	0.83	24.92	0.90	60.21	0.79	13.14	**0.61**	4.28	0.55	**6.94**
	CAMIA (LR/PCA)	**0.81**	**32.89**	**0.95**	**72.22**	**0.91**	**64.57**	**0.82**	**26.72**	0.60	**4.46**	0.54	2.70
6.9B	LOSS	0.78	15.14	0.92	26.35	0.87	33.88	0.78	16.51	0.60	1.85	0.57	6.61
	Zlib	0.78	13.17	0.80	12.38	0.90	38.46	0.77	13.26	0.59	2.78	0.55	7.54
	Min-K%	0.75	20.37	0.92	60.79	0.87	34.95	0.78	18.98	0.59	2.10	0.57	6.20
	Min-K%++	0.65	4.40	0.73	17.78	0.84	25.48	0.67	8.55	0.58	1.92	0.56	5.13
	Reference	0.72	8.29	0.46	2.06	0.64	0.64	0.60	1.31	0.58	1.77	**0.64**	9.87
	CAMIA (Edg.)	**0.82**	28.69	0.86	29.05	0.90	55.32	0.79	11.89	**0.61**	**6.45**	0.58	**10.01**
	CAMIA (LR/PCA)	**0.82**	**33.23**	**0.95**	**70.79**	**0.91**	**63.72**	**0.82**	**24.53**	**0.61**	5.12	0.57	4.31
12B	LOSS	0.79	15.03	0.92	17.30	0.88	35.05	0.77	16.54	0.61	2.14	0.58	7.14
	Zlib	0.78	14.86	0.81	9.37	0.91	36.70	0.77	11.77	0.60	3.07	0.56	8.57
	Min-K%	0.77	21.66	0.92	51.11	0.88	35.21	0.78	20.99	0.60	2.43	0.58	6.49
	Min-K%++	0.68	6.31	0.70	22.70	0.86	27.23	0.67	9.80	0.59	1.96	0.58	6.51
	Reference	0.73	8.74	0.45	0.48	0.61	0.69	0.58	1.05	0.61	2.72	0.67	10.57
	CAMIA (Edg.)	**0.82**	28.06	0.85	27.62	0.91	61.38	0.79	11.77	**0.61**	**6.95**	**0.59**	**10.66**

Continued on next page

Size	Attack	ArXiv		Mathematics		GitHub		PubMed		HackerNews		Pile-CC	
		AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
	CAMIA (LR/PCA)	**0.82**	**36.06**	**0.95**	**69.84**	**0.92**	**63.78**	**0.82**	**21.28**	**0.61**	5.74	0.58	4.89

Table A.2: Original CAMIA results for GPT-Neo suite (11)

Size	Attack	ArXiv		Mathematics		GitHub		PubMed		HackerNews		Pile-CC	
		AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
	LOSS	0.76	9.40	0.95	77.94	0.83	24.84	0.81	19.53	0.57	1.59	0.53	2.29
	Zlib	0.76	6.94	0.82	27.62	0.86	39.68	0.79	21.48	0.57	2.25	0.51	2.53
	Min-K%	0.72	9.11	0.94	75.24	0.82	25.43	0.80	21.13	0.56	1.52	0.53	2.61
125M	Min-K%++	0.62	3.91	0.70	16.35	0.78	18.78	0.67	10.29	0.54	2.14	0.52	2.13
	Reference	0.65	2.91	0.56	0.32	0.67	1.12	0.73	6.40	0.51	0.84	0.51	2.29
	CAMIA (Edg.)	0.81	23.09	0.88	16.51	0.87	50.53	0.82	19.94	0.59	2.41	0.54	4.10
	CAMIA (LR/PCA)	**0.82**	**28.00**	**0.95**	**77.30**	**0.89**	**65.85**	**0.84**	**28.90**	**0.57**	**3.47**	**0.52**	**1.50**
	LOSS	0.78	12.74	0.93	63.02	0.86	39.10	0.80	18.81	0.59	1.74	0.54	4.56
	Zlib	0.78	11.91	0.80	14.76	0.88	51.28	0.78	19.48	0.58	2.19	0.53	4.24
	Min-K%	0.75	15.09	0.93	70.48	0.86	38.94	0.80	22.24	0.57	1.96	0.54	4.10
1.3B	Min-K%++	0.66	4.71	0.71	26.03	0.81	33.35	0.68	9.01	0.56	1.88	0.53	2.99
	Reference	0.71	5.86	0.49	1.43	0.66	1.97	0.70	1.31	0.52	1.66	0.55	4.60
	CAMIA (Edg.)	0.82	25.80	0.83	18.73	0.90	62.50	0.81	17.70	0.60	4.17	0.55	6.16
	CAMIA (LR/PCA)	**0.82**	**32.23**	**0.95**	**74.44**	**0.91**	**65.96**	**0.83**	**27.94**	**0.59**	**3.75**	**0.54**	**2.80**
	LOSS	0.79	16.51	0.93	55.71	0.87	41.86	0.80	21.25	0.59	1.61	0.55	4.97
	Zlib	0.78	14.40	0.81	15.87	0.89	50.32	0.78	18.63	0.58	2.43	0.54	5.34
	Min-K%	0.76	21.09	0.93	69.68	0.87	42.18	0.80	23.46	0.57	1.79	0.55	4.64
2.7B	Min-K%++	0.66	7.91	0.72	27.14	0.83	34.20	0.69	12.18	0.57	1.96	0.54	3.79
	Reference	0.72	7.06	0.52	0.32	0.65	1.54	0.69	1.66	0.52	1.88	0.57	5.60
	CAMIA (Edg.)	0.82	28.57	0.86	21.75	0.91	60.59	0.81	15.84	0.59	2.69	0.56	5.97
	CAMIA (LR/PCA)	**0.83**	**37.03**	**0.95**	**73.65**	**0.92**	**67.50**	**0.83**	**23.63**	**0.58**	**4.13**	**0.55**	**3.23**