DIANA: CALIBRATING THE PER-TOKEN NULL FOR MEMBERSHIP INFERENCE ON FINE-TUNED LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Membership inference attacks on language models detect whether a specific text was included in the training data. Fine-tuning on a domain complicates this task: an increase in a token's log-probability over the pre-trained model, which we call its gain, may reflect domain-level learning rather than memorization of the target sequence itself, inflating false positives in reference-based attacks. We introduce DIANA (**D**istribution-**I**nformed **A**ttacks with **N**ull **A**djustment), a plug-and-play calibration that isolates memorization from domain adaptation. Our two key observations are: (i) for non-members, the target model's own next-token distribution approximates the domain distribution. This motivates centering: subtracting the target-weighted mean gain over possible next tokens from the observed gain. (ii) Under our modeling assumptions, the centered gain has zero conditional mean for non-members and positive conditional mean for members. To account for differences in background variability, DIANA divides the centered gain by the regularized gain variance to obtain a calibrated gain. Hence, DIANA requires only the log-probabilities of the target and pre-trained models—no auxiliary data or additional training—and integrates with any reference-based attacks, which then aggregate the calibrated gains under their original rules. Across six corpora, RATIO+DIANA achieves $4.2\times$ the mean true-positive rate of RATIO at a 0.1% false-positive rate and outperforms all evaluated prior attacks on every corpus.

1 INTRODUCTION

Membership inference on fine-tuned language models determines whether a candidate sequence belongs to the fine-tuning set (Shokri et al., 2017). We consider an adversary with query access to a fine-tuned target $\mathcal{M}^{\mathrm{T}}$ and its pre-trained reference $\mathcal{M}^{\mathrm{R}}$, but no access to the fine-tuning data or knowledge of its distribution. Fine-tuning can increase a sequence's likelihood through both memorization of individual training examples (Carlini et al., 2021; Mireshghallah et al., 2022b) and adaptation to patterns shared across the domain. For instance, tutorial language can become more predictable even in documents not in training (Figure 1). A likelihood improvement therefore provides ambiguous membership evidence: it may reflect exposure to the candidate sequence or generalization from other sequences in the same domain. Distinguishing these effects requires a baseline for the domain improvement expected without membership.

Reference-based attacks, such as DIFF and RATIO, calibrate for sequence difficulty by comparing target and reference likelihoods (Carlini et al., 2021; Watson et al., 2021). Their scores can be expressed in terms of per-token *gains*, the differences between target and reference log-probabilities. With auxiliary in-domain data, shadow-model attacks estimate a non-member baseline by training models on datasets that exclude the candidate sequence (Ye et al., 2022; Carlini et al., 2022; Hayes et al., 2025). Without such auxiliary data, Window-Based Comparison (WBC) (Chen et al., 2026) limits the influence of extreme gains through windowed sign comparisons, while EZ-MIA (Ilić et al., 2026) selects positions where the target's top prediction is incorrect. These rules change how gains are selected or aggregated, but do not estimate the gain expected from domain adaptation at each prefix. A positive gain can therefore count as membership evidence even when it is typical of non-members.

Our approach. We introduce DIANA (Distribution-Informed Attacks with Null Adjustment), a calibration for reference-based membership inference that estimates a token-level non-member baseline from the target and reference predictions, without auxiliary data or additional training. At

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

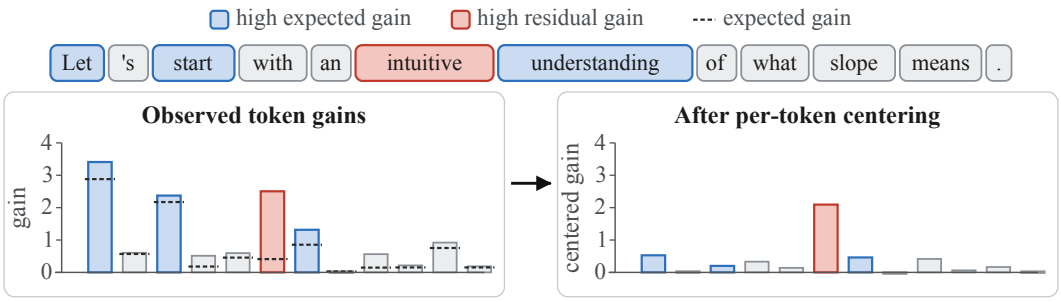


Figure 1: **Recurring tutorial language can mimic membership evidence.** In 10,000 held-out Cosmopedia-Khan Academy documents (Ben Allal et al., 2024), *Let's start* appears in 17.5% and *understanding* in 67.2% (case-insensitive counts). *Left:* observed gains in a member sentence; dashed lines show the expected gain over possible next tokens under the target at each prefix. *Right:* subtracting that baseline from every token, before variance weighting. The gain on *Let* falls from 3.41 to 0.53 nats, while *intuitive* retains 2.09 of its 2.50 nats, a larger excess over its local baseline. DIANA computes these baselines from model outputs, without corpus-frequency information.

each prefix, DIANA computes the target-to-reference log-probability gain across the vocabulary and subtracts its target-weighted mean from the observed token's gain. The target's predictive distribution provides a proxy for domain continuations, but memorization can raise both the observed gain and the baseline being subtracted. We therefore analyze both cases, establishing conditions under which the resulting *centered gain* has zero conditional mean for non-members (Proposition 1) and strictly positive conditional mean for members (Theorem 1); Figure 1 shows a representative real-world illustration of the centering operation. DIANA then divides the centered gain by the target-weighted gain variance plus a positive hyperparameter τ, down-weighting prefixes with high gain variability. The hyperparameter acts as a variance floor, bounding the weight at low variance (Equation (5)). We fix $\tau = 1$, supported by stable AUC across tested values between 0.25 and 4 (Appendix I). DIANA is a plug-and-play calibration for attacks such as DIFF and RATIO, as it replaces their per-token gains with calibrated gains, leaving the rest of the scoring rules unchanged. The calibration just requires full next-token distributions from $\mathcal{M}^{\mathrm{T}}$ and $\mathcal{M}^{\mathrm{R}}$, available in one forward pass per model.

Across all six corpora in our primary OLMOE-1B-7B evaluation, RATIO+DIANA improves AUC and true-positive rate (TPR) over RATIO, raising mean TPR at 0.1% false-positive rate (FPR) from 11.2% to 47.4% (Table 1 and appendix E). AutoMathText illustrates one of the largest relative gains: TPR rises from 2.5% to 35.4% (14.2×). Ablations show that centering supplies most of the improvement, with variance weighting providing further gains (Table 3, Appendix G). Further evaluations span five additional model families, three extra corpora, and diverse training and access settings (Appendices F and K).

2 PRELIMINARIES

Threat model. Let $M \in \{0, 1\}$ indicate whether a target sequence $\mathbf{x}$ belongs to the private fine-tuning set of a target model $\mathcal{M}^{\mathrm{T}}$. The adversary seeks to infer M using query access to the target model and its pre-trained base $\mathcal{M}^{\mathrm{R}}$, as in prior fine-tuning MIAs (Chen et al., 2026; Ilić et al., 2026). The adversary has no access to the fine-tuning data or its distribution, to gradients, or to model parameters. We consider two levels of output access. (i) *Full-distribution access*: the adversary obtains both models' complete next-token distributions at every prefix of the target sequence, the output access also assumed by Min-K%++ (Zhang et al., 2025). For released fine-tuned checkpoints, these probabilities come from ordinary forward passes; a hosted target supports this setting when its interface returns full-vocabulary probabilities. Our main results use this setting. (ii) *Top-k probability access*: at each prefix, the adversary obtains both models' probabilities only for the observed token and for the union of the two models' k most likely next tokens. In this setting, DIANA approximates the mean and variance on that union. Our evaluation shows gains even at $k = 5$ (Section 4.2 and appendix K).

Reference-based membership inference. Let $\mathbf{x} = (x_1, \ldots, x_n)$ be a target text sequence, and $\mathcal{V} = \{1, \ldots, V\}$ be the shared vocabulary. At each scored position $j \in \{2, \ldots, n\}$, the prefix is $x_{<j} = (x_1, \ldots, x_{j-1})$ and the observed next token is x_j. Write $p_j^{\mathrm{T}}(v) \coloneqq p^{\mathrm{T}}(v \mid x_{<j})$ and

$p_j^{\mathrm{R}}(v) \coloneqq p^{\mathrm{R}}(v \mid x_{<j})$ for the next-token probability mass functions of the fine-tuned target $\mathcal{M}^{\mathrm{T}}$ and the pre-trained reference $\mathcal{M}^{\mathrm{R}}$, respectively. Both models use the same tokenizer and assign positive probability to every token. Their *conditional log-gain function* is

$$G_j(v) \coloneqq \log \frac{p_j^{\mathrm{T}}(v)}{p_j^{\mathrm{R}}(v)} = \log p_j^{\mathrm{T}}(v) - \log p_j^{\mathrm{R}}(v), \qquad v \in \mathcal{V}. \tag{1}$$

All logarithms are natural. At a fixed prefix, $(G_j(v))_{v\in\mathcal{V}}$ records how fine-tuning changes the log-probability of every possible next token.

Existing reference-based attacks such as DIFF and RATIO (Carlini et al., 2021; Watson et al., 2021) use this function at the observed next token:

$$G_j^{\mathrm{obs}} \coloneqq G_j(x_j) = \log p_j^{\mathrm{T}}(x_j) - \log p_j^{\mathrm{R}}(x_j). \tag{2}$$

Write $\ell^{\mathrm{T}}(\mathbf{x}) \coloneqq -\frac{1}{n-1}\sum_{j=2}^{n} \log p_j^{\mathrm{T}}(x_j)$ for the target's mean negative log-likelihood, and define the reference loss $\ell^{\mathrm{R}}(\mathbf{x})$ analogously. DIFF uses the sentence-level loss reduction:

$$\ell^{\mathrm{R}} - \ell^{\mathrm{T}} = \frac{1}{n-1}\sum_{j=2}^{n}\left(\log p_j^{\mathrm{T}}(x_j) - \log p_j^{\mathrm{R}}(x_j)\right) = \frac{1}{n-1}\sum_{j=2}^{n} G_j^{\mathrm{obs}}. \tag{3}$$

RATIO assumes that a larger reduction gives stronger evidence of membership, and uses the sentence-level loss ratio, with smaller values giving stronger evidence. This can be expressed as follows:

$$\frac{\ell^{\mathrm{T}}}{\ell^{\mathrm{R}}} = \frac{\ell^{\mathrm{R}} - (\ell^{\mathrm{R}} - \ell^{\mathrm{T}})}{\ell^{\mathrm{R}}} = 1 - \frac{\ell^{\mathrm{R}} - \ell^{\mathrm{T}}}{\ell^{\mathrm{R}}} = 1 - \frac{1}{(n-1)\ell^{\mathrm{R}}}\sum_{j=2}^{n} G_j^{\mathrm{obs}}. \tag{4}$$

Signal model. A positive observed gain G_j^{obs} means that the target model has lower loss than the reference model at the j-th position of the sequence $\mathbf{x}$. Intuitively, this is evidence that $\mathbf{x}$ is a member when training the target model. One might naturally assume that the larger G_j^{obs} is, the more confidence one should have regarding the membership. A key observation in Chen et al. (2026) is that this assumption is actually incorrect due to the **domain adaptation** effect. A large G_j^{obs} can occur only when the fine-tuning process encounters $x_{\le j} = (x_1, \ldots, x_j)$ or similar sequences many times, which can happen only when similar patterns appear in many instances. In that case, however, this significant loss reduction is not a good signal for membership of $\mathbf{x}$ because it will happen whether $\mathbf{x}$ is a member or not. For example, in Python code, `self` commonly follows `def __init__(`. A large gain on this token therefore does not indicate membership of any individual sequence. In other words, domain adaptation has a confounding effect when using G_j^{obs} as membership signal. Membership of $\mathbf{x}$ manifest not as large gain in one or a few positions, but as consistent small gain over many positions.

In Chen et al. (2026), the impact of this domain adaptation effect is mitigated by computing the sign of the average G_j^{obs} within each small window of multiple positions and combining the signs across many overlapping windows to determine membership. Since tokens affected by domain adaptation are expected to occur only rarely, they influence only a small fraction of the windows. This approach, however, does not directly address the domain adaptation effect itself, but instead reduces its adverse impact through aggregation. In this paper, we show how to directly remove the domain adaptation effect, yielding a cleaner membership signal than G_j^{obs}.

3 DIANA: DISTRIBUTION-INFORMED ATTACKS WITH NULL ADJUSTMENT

To distinguish memorization from domain adaptation, we need a baseline: how much gain a token would receive if the target sequence had not been used for training. Without access to the fine-tuning data, this baseline cannot be estimated with shadow models, which are trained on in-domain data that excludes the target sequence (Carlini et al., 2022; Ye et al., 2022). DIANA instead estimates baseline from the target and reference predictions at each prefix. Section 3.1 defines the resulting calibrated gain, and Section 3.2 shows when it separates members from non-members.

3.1 THE CALIBRATED GAIN

We now define DIANA's calibrated gain $\tilde{s}_j$ and explain the role of each term. To obtain the calibrated gain $\tilde{s}_j$, we start with $G_j^{\text{obs}} = G_j(x_j)$, first subtract from it the target-weighted mean gain, and then normalize it by dividing by the target-weighted variance plus a hyperparameter $\tau > 0$:

$$\tilde{s}_j \coloneqq \frac{G_j^{\text{obs}} - \sum_{v \in \mathcal{V}} p_j^{\text{T}}(v) G_j(v)}{\tau + \sum_{v \in \mathcal{V}} p_j^{\text{T}}(v) \left[G_j(v) - \sum_{u \in \mathcal{V}} p_j^{\text{T}}(u) G_j(u) \right]^2} = \frac{s_j}{\tau + \sigma_j^2}, \tag{5}$$

$$\mu_j \coloneqq \mathbb{E}_{W \sim p_j^{\text{T}}}[G_j(W)], \qquad \sigma_j^2 \coloneqq \text{Var}_{W \sim p_j^{\text{T}}}(G_j(W)), \qquad s_j \coloneqq G_j^{\text{obs}} - \mu_j.$$

Here μ_j is the target-weighted mean and and σ_j^2 is the variance of the gain over possible next tokens. We call subtracting μ_j from the observed gain as *centering*, which results in the centered gain s_j. The sign of s_j indicates whether the observed gain is above or below the target-weighted mean. The variance σ_j^2 measures the spread of possible next-token gains. For a non-member, the observed token is essentially one draw from the next-token distribution:the same positive centered gain is weaker evidence of membership where the variance is large, and dividing by $\tau + \sigma_j^2$ down-weights such prefixes. The hyperparameter $\tau > 0$ keeps the weight bounded when the variance is almost zero. It is fixed across all target sequences and positions, and results are not sensitive to its value (Section 4.2 and appendix I).

DIANA's calibrated gain $\tilde{s}_j$ can be used to replace G_j^{obs} in a reference-based attack such as DIFF or RATIO (Equations (3) and (4)), leaving the rest of the attack unchanged; we refer to the results as DIFF+DIANA and RATIO+DIANA. Appendix B gives the pseudocode, which needs one forward pass of each model over the target sequence. Section 3.2 shows when the calibrated gain separates members from non-members in expectation, including how τ keeps the member mean positive.

3.2 MEAN SEPARATION OF THE CALIBRATED GAIN

We now show that, under the conditions below, the calibrated gain $\tilde{s}_j$ has zero expectation for a non-member and positive expectation for a member. This helps make $\tilde{s}_j$ a better membership signal than G_j^{obs}. Let $\mathcal{D}$ denote the distribution of texts in the fine-tuning domain. Given the target sequence's prefix $x_{<j}$, write $p_j^{\mathcal{D}}(v)$ for the probability that a continuation drawn from this domain has next token v. Formally, $p_j^{\mathcal{D}}(v) \coloneqq \Pr_{\mathbf{X} \sim \mathcal{D}}(X_j = v \mid X_{<j} = x_{<j})$. This describes how text in the domain continues the prefix, independently of whether the target sequence was included in training.

Different fine-tuning datasets and training runs can produce different model predictions. Write p_j^- and p_j^+ for the next-token predictions averaged over these choices under the same training procedure when training excludes or includes the target sequence, respectively: $p_j^-(v) = \mathbb{E}_{\text{train}}[p_j^{\text{T}}(v) \mid M = 0]$ and $p_j^+(v) = \mathbb{E}_{\text{train}}[p_j^{\text{T}}(v) \mid M = 1]$, with the target sequence and prefix fixed. A queried target model supplies one realization of the corresponding training case. For this analysis, we assume that the domain and reference distributions assign positive probability to every token in a vocabulary with at least two tokens.

Non-members. Fine-tuning with domain data can make p_j^- differ substantially from the pre-trained p_j^{R}. If the model learns the domain well and excluding one target sequence has little effect on that learning, p_j^- should approximate $p_j^{\mathcal{D}}$. We analyze the simplifying scenario where $p_j^- = p_j^{\mathcal{D}}$. For a next token drawn from this domain, the observed gain and the mean subtracted during centering then have the same expectation.

Proposition 1 (Exact conditional null centering)**.** *For a target sequence* $\mathbf{x}$ *not used in training* $\mathcal{M}^{\text{T}}$*, under the assumption* $p_j^- = p_j^{\mathcal{D}}$*, every scored position* $j \in \{2, \ldots, n\}$ *satisfy*

$$\mathbb{E}[\tilde{s}_j] = 0, \tag{6}$$

where $\tilde{s}_j$ *is as defined in Equation* (5)*.*

Proof sketch. The next token follows $p_j^{\mathcal{D}}$, while centering subtracts the expected gain under p_j^-. The numerator of $\tilde{s}_j$ therefore has mean $\sum_v (p_j^{\mathcal{D}}(v) - p_j^-(v))G_j(v)$, which vanishes when $p_j^- = p_j^{\mathcal{D}}$. The denominator $\tau + \sigma_j^2$ is constant at this prefix, so $\mathbb{E}[\tilde{s}_j] = 0$. Appendix A.1 gives the full proof. □

Section 3.3 empirically tests on Proposition 1 on held-out text (Figure 2b, Cablibrated plot).

Members. We use a *single-spike model* for a member token: memorizing the target sequence increases the probability of its observed token x_j by mixing the domain prediction with a point mass at that token,

$$p_j^+(v) = (1 - \alpha_j)p_j^{\mathcal{D}}(v) + \alpha_j \mathbf{1}_{\{v = x_j\}}, \qquad 0 < \alpha_j < 1, \quad v \in \mathcal{V}, \tag{7}$$

where α_j is the memorization strength.

Theorem 1 (Positive conditional member mean)**.** *For a target sequence* $\mathbf{x}$ *included in training* $\mathcal{M}^{\mathrm{T}}$*,* $\exists \tau > 0$ *such that under the member next-token distribution* p_j^+ *in Equation* (7)*, each scored position* $j \in \{2, \ldots, n\}$ *satisfies*

$$\mathbb{E}[\tilde{s}_j] > 0, \tag{8}$$

where $\tilde{s}_j$ *is as defined in Equation* (5)*.*

Proof sketch. Equation (7) separates the numerator of $\tilde{s}_j$ into a zero-mean baseline deviation and a strictly positive average gain due to memorization. Once τ is large enough, $\mathbb{E}[\tilde{s}_j] > 0$. Appendix A.2 gives the full proof, an explicit τ threshold that guarantees $\mathbb{E}[\tilde{s}_j] > 0$, and an example showing why this threshold is needed. □

Behavior near memorization saturation. Theorem 1 guarantees a positive member mean but does not say how it changes as memorization strengthens. As memorization approaches saturation, the model assigns almost all probability to the memorized token. The mean subtracted during centering then approaches that token's observed gain, driving the centered gain back toward zero. The expansion in the proof of Theorem 1 makes this shape precise.

Proposition 2 (Concavity in memorization strength)**.** *Under the single-spike assumptions of Theorem 1, the expected centered member gain* $\mathbb{E}[s_j]$ *is strictly concave in* α_j *(memorization strength).*

Proof sketch. After the baseline term averages to zero, each remaining memorization contribution is strictly concave in α_j. Their average under the fixed domain distribution is therefore strictly concave as well. Appendix A.3 gives the full proof. □

Section 3.3 confirms Proposition 2 experimentally (Figure 2a), empirically showing the concavity of $\mathbb{E}[s_j]$ over the training data.

3.3 Empirical Validation of Theoretical Results

We check two predictions of the analysis on Cosmopedia-Stanford Ben Allal et al. (2024) with OLMoE Muennighoff et al. (2025) (Figure 2). Panel (a) asks whether the member mean rises and then falls as memorization strengthens (Proposition 2). Panel (b) asks whether the target-weighted mean gain predicts the gain on non-member tokens, so that calibration leaves a mean near zero (Proposition 1). These moments come from a trained target rather than the idealized prediction p_j^-, so we write $\hat{\sigma}_j^2$ for the variance computed from the target.

Members (a). Memorization strength is not observable, so we use the target's mean probability on member tokens as a proxy; it increases throughout training. We track 800 member sequences per seed for twenty epochs with three seeds. The mean centered gain rises, peaks at epoch six, and then declines in all three seeds, and the theoretical shape fits this trajectory with $R^2 = 0.952$. The calibrated member mean with $\tau = 1$ follows the same pattern: it peaks at epoch four and remains positive at every checkpoint, consistent with Theorem 1. Appendix N gives the fit and checkpoint protocol.

Non-members (b). We score 5,000 held-out sequences with the one-epoch target and group their tokens by $\hat{\sigma}_j^2$. This grouping shows where the domain adaptation described in Section 2 inflates non-member gains. The variance $\hat{\sigma}_j^2$ is zero when the target and reference predict the same next-token distribution, and it grows as fine-tuning changes the log-probabilities of likely tokens by different amounts. Because these sequences were not in the training set, such changes come from fine-tuning on other text in the domain; the red dashed curve shows that μ_j, which equals $D_{\mathrm{KL}}(p_j^{\mathrm{T}} \| p_j^{\mathrm{R}})$ (Appendix H) and is another measure of this change, rises with $\hat{\sigma}_j^2$.

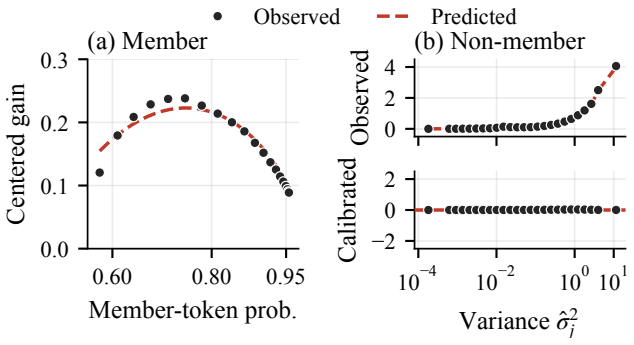


Figure 2: **Checking the predictions on Stanford.** (a) Mean centered member gain at each training checkpoint; the dashed curve is the fitted theoretical shape. (b) Non-member tokens grouped by computed variance. Top: observed gain and target-weighted mean gain μ_j (dashed). Bottom: calibrated gain with $\tau = 1$ and its predicted mean of zero.

At high-variance prefixes, the observed gain is large even though these sequences were never trained on: the mean gain rises from nearly zero to 4.06 nats in the highest-variance group, and tokens with $\hat{\sigma}_j^2 > 1$, 16.5% of the total, carry 50.7% of all observed non-member gain. An uncalibrated attack counts these gains as membership evidence. Proposition 1 predicts that μ_j matches the observed gain at every prefix. In the top plot, the dashed curve (mean μ_j) follows the points across the whole range, accounting for 90.6–98.8% of the observed gain in each group and 93.3% overall (0.400 of 0.429 nats). The bottom plot applies the full calibration $\tilde{s}_j = s_j/(1 + \hat{\sigma}_j^2)$ at each token before averaging, on a y-axis with the same span. Proposition 1 predicts a zero mean in every group, and the measured means stay within 0.033 of zero in all groups; in the highest-variance group, the 4.06-nat observed gain becomes a calibrated mean of 0.003. The overall calibrated mean is 0.016 (95% interval $[0.0155, 0.0167]$), small but positive, because μ_j slightly underestimates the observed gain; dividing by $1 + \hat{\sigma}_j^2$ shrinks this remainder, especially at high-variance prefixes. Appendix M repeats the non-member check on all six corpora, verifies that members retain higher centered means, and checks the variance used for calibration.

4 Experiments

We evaluate DIANA across attack rules, model families, and fine-tuning domains. Our primary comparison uses OLMoE-1B-7B (Muennighoff et al., 2025) on the six Cosmopedia corpora evaluated by WBC (Chen et al., 2026). We choose its LLM-generated texts to reduce the risk of pre-training contamination when measuring fine-tuning membership (Ben Allal et al., 2024). More experiments cover Llama-3.1 (Grattafiori et al., 2024), PYTHIA (Biderman et al., 2023), OPT (Zhang et al., 2022b), Qwen2.5 (Qwen Team, 2024), and Mamba (Gu & Dao, 2024), plus three additional non-generated corpora: XSum (Narayan et al., 2018), CC-News (Hamborg et al., 2017), and Wikipedia (Wikimedia Foundation, 2023). We compare reference-free, white-box, and reference-based attacks, then isolate the contributions of centering and variance weighting and test sensitivity to text length, τ, and probability access. We further vary the target size (PYTHIA-160M to 6.9B), the reference for a fixed target, training duration, and LoRA fine-tuning (Section 4.2).

Setup. We fine-tune each primary OLMoE target for one epoch with learning rate at 5×10^{-6}, using its pre-trained base as reference. We partition each corpus and evaluate on 20k members and 20k non-members (10k per class for Khan), tokenized to 512-token records. We report AUC and the true-positive rate (TPR), the fraction of members detected, at false-positive rates (FPR), the fraction of non-members wrongly flagged, of 10%, 1%, and 0.1% (Carlini et al., 2022). All evaluations use Equation (5) with $\tau = 1$, except the τ sweep and the $\tau = 0$ ablation. Appendices D and L give the protocols and model-utility checks.

Defense. We evaluate DP-SGD (Abadi et al., 2016) on PYTHIA-410M fine-tuned on Khan Academy (Appendix F, Table 14). At $\varepsilon \approx 8$ and 4, DIANA and every compared attack are near chance (AUC

Table 1: **Membership inference on three OLMoE fine-tunes**, using the same Cosmopedia corpora as Chen et al. (2026). Best bold; second best underlined. Access: sequence likelihoods (Seq.), observed-token probabilities (Token), Token plus the target's top prediction (Top-1), full distributions (Vocab.), model weights (Weights), or gradients (Grad.). Seq.+M additionally uses known members; Bag-of-Words is a supervised control that uses only the text.

Method	Access	Stanford				AutoMathText				WikiHow			
		AUC	TPR at FPR			AUC	TPR at FPR			AUC	TPR at FPR		
			10%	1%	**0.1%**		10%	1%	**0.1%**		10%	1%	**0.1%**
Text-only control													
Bag-of-Words (Das et al., 2025)	Text	.501	.101	.011	.001	.498	.102	.010	.001	.503	.102	.010	.001
Reference-free													
Loss (Yeom et al., 2018)	Seq.	.674	.270	.070	.012	.609	.174	.023	.003	.689	.279	.058	.012
ZLIB (Carlini et al., 2021)	Seq.	.608	.190	.030	.006	.589	.144	.017	.002	.628	.201	.026	.002
Lowercase (Carlini et al., 2021)	Seq.	.680	.281	.054	.013	.609	.179	.028	.002	.698	.299	.052	.001
Min-K% (Shi et al., 2023)	Token	.702	.309	.098	.025	.647	.208	.031	.004	.723	.339	.077	.021
Min-K%++ (Zhang et al., 2025)	Vocab.	.805	.531	.224	.080	.747	.374	.072	.013	.798	.501	.185	.053
ReCall (Xie et al., 2024)	Seq.	.677	.248	.038	.010	.648	.214	.032	.005	.669	.249	.035	.006
Neighborhood (Mattern et al., 2023)	Seq.	.574	.146	.015	.002	.565	.146	.017	.002	.559	.129	.014	.001
Con-ReCall (Wang et al., 2025)	Seq.+M	.505	.125	.013	.003	.496	.116	.017	.002	.474	.105	.015	.002
DC-PDD (Zhang et al., 2024)	Token	.653	.220	.025	.001	.647	.216	.030	.000	.660	.220	.001	.001
White-box													
MoPe (Li et al., 2023)	Weights	.802	.514	.154	.023	.759	.460	.158	.029	.804	.518	.186	.036
GradNorm (Nasr et al., 2019)	Grad.	.780	.465	.141	.019	.633	.234	.032	.005	.736	.414	.106	.021
Reference-based													
Diff (Watson et al., 2021)	Seq.	.810	.533	.235	.093	.757	.452	.159	.006	.808	.519	.175	.055
Diff + Diana	Vocab.	.918	.799	.623	.470	.886	.722	.462	.254	.904	.775	.576	.406
EZ-MIA (Ilić et al., 2026)	Top-1	.894	.755	.542	.367	.873	.706	.467	.285	.891	.744	.494	.284
EZ-MIA + Diana	Vocab.	.906	.778	.602	.447	.880	.722	.493	.308	.897	.761	.563	.395
WBC (Chen et al., 2026)	Token	.894	.746	.514	.341	.840	.618	.343	.170	.876	.696	.406	.204
WBC + Diana	Vocab.	.919	.808	.630	.488	.891	.739	.483	.288	.900	.772	.555	.354
Ratio (Carlini et al., 2021)	Seq.	.876	.690	.380	.146	.842	.606	.211	.025	.830	.559	.171	.049
Ratio + Diana	Vocab.	**.925**	**.824**	**.678**	**.542**	**.900**	**.764**	**.551**	**.354**	**.911**	**.793**	**.618**	**.473**

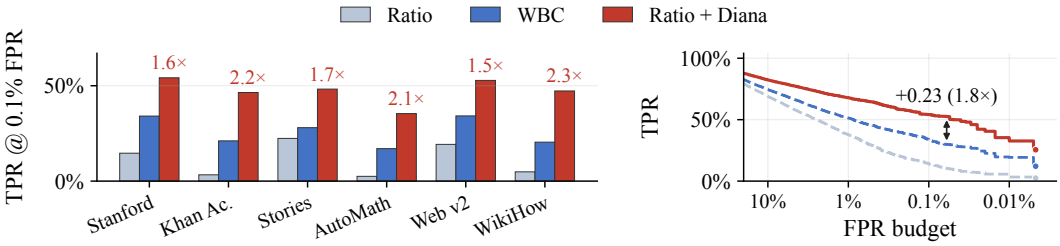


Figure 3: **Detection at low false-positive rates.** Baselines: Ratio (Carlini et al., 2021) and WBC (Chen et al., 2026). *Left:* Ratio+Diana detects 1.5–2.3× as many members as WBC at 0.1% FPR across six corpora. *Right:* Stanford ROC; the arrow marks the gap to WBC at 0.06% FPR, and right-edge circles mark the zero-FPR TPR (25.5% with Diana versus 12.1% for WBC).

at most 0.505), but fine-tuning then reduces held-out perplexity by only 4.8% and 3.4%, compared with 35.5% without DP (Appendix L): privacy budgets strong enough to stop Diana remove most of the model's utility.

Table 2: **Transfer across targets and corpora.** DIANA improves RATIO in both AUC and TPR at 1% FPR for every target and corpus. Each target uses its pre-trained base as reference.

	AUC		TPR at 1% FPR	
Target / corpus	RATIO	+DIANA	RATIO	+DIANA
Stanford: target model				
Llama-3.1-8B	.939	**.960**	.696	**.851**
PYTHIA-2.8B	.765	**.864**	.099	**.406**
OPT-1.3B	.708	**.837**	.058	**.302**
Qwen2.5-1.5B	.709	**.780**	.045	**.096**
Mamba-1.4B	.735	**.859**	.055	**.276**
OLMOE: *non-generated corpus*				
XSum	.917	**.967**	.021	**.835**
CC-News	.875	**.928**	.017	**.039**
Wikipedia	.926	**.972**	.251	**.879**

Table 3: **Gain ablations.** Removing centering, variance weighting, or the variance floor ($\tau = 0$) lowers both AUC and TPR. TPR is measured at 0.1% FPR; $\tau = 1$ when used.

	Components			Detection	
Variant	$-\mu_j$	w_j	$\tau > 0$	AUC	TPR
Components and τ					
Observed gain	✗	✗	✗	.854	.112
Weighting only	✗	✓	✓	.873	.110
Centered gain	✓	✗	✗	.901	.418
$\tau = 0$	✓	✓	✗	.827	.002
DIANA	✓	✓	✓	**.911**	**.474**
Weight denominator in Equation (5)					
$\tau + \mu_j$	✓	✓	✓	.909	.466
$\tau + \sigma_j$	✓	✓	✓	.909	.463
$\sqrt{\tau + \sigma_j^2}$	✓	✓	✓	.909	.467
$\tau + \sigma_j^2$	✓	✓	✓	**.911**	**.474**

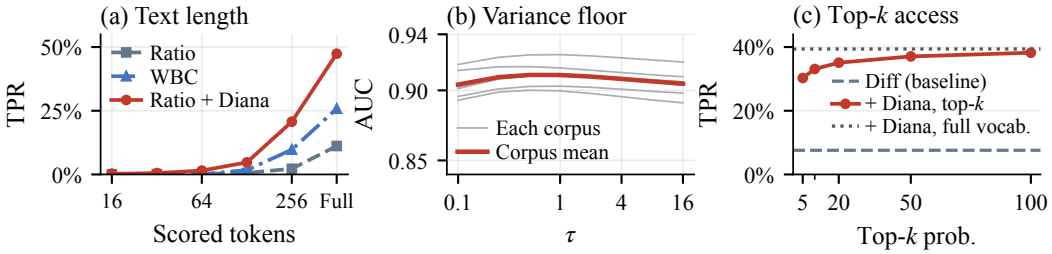


Figure 4: **Sensitivity to text length, τ, and probability access.** (a) Longer target sequences strengthen detection. (b) RATIO+DIANA has a broad AUC plateau across values of τ. (c) Top-50 moments approach full-vocabulary detection with DIFF calibration. Panels (a,c) show six-corpus mean TPR at 0.1% FPR; "Full" in (a) is 512 tokens.

4.1 MEMBERSHIP INFERENCE PERFORMANCE

Table 1 compares attacks on the same complete target-sequence splits for three representative corpora; Appendix E gives all six with uncertainty. DIANA improves AUC and TPR at 0.1% FPR for DIFF, RATIO, and WBC on every corpus. RATIO+DIANA leads all four reported metrics, raising mean TPR at 0.1% FPR from 0.112 to 0.474. Applied to EZ-MIA (Ilić et al., 2026), it also raises mean TPR from 0.323 to 0.393, with gains on all six corpora. Calibration also reverses the ranking of the host attacks. Before calibration, RATIO detects fewer members than WBC (0.112 versus 0.258 mean TPR at 0.1% FPR), because averaging raw gains lets domain-adapted tokens dominate the score. After calibration, RATIO+DIANA (0.474) exceeds WBC +DIANA (0.397): once each token's domain baseline is removed, averaging the full magnitude of the evidence outperforms WBC's robust sign test. Gains are largest where uncalibrated RATIO is weakest, on Khan Academy and AutoMathText (Appendix E).

Figure 3 complements these fixed operating points with all six corpora and the Stanford ROC. The gains over RATIO and WBC persist as the FPR budget tightens, with RATIO+DIANA detecting 25.5% of members before the first false positive, versus 12.1% for WBC.

4.2 TRANSFER, ABLATIONS, AND SENSITIVITY

Transfer across models and corpora. Calibrating RATIO improves both AUC and TPR at 1% FPR for every target and corpus in Table 2. At 1% FPR, TPR increases 4.1× on PYTHIA-2.8B, 5.2× on OPT-1.3B, and 5.0× on Mamba-1.4B; these one-epoch targets reduce held-out perplexity

by 35.1–37.0% (Appendix L). The smallest gain among targets is on Qwen2.5-1.5B, from 0.045 to 0.096; its fine-tuning also changes the model least (held-out perplexity falls by only 4.9%), leaving little domain adaptation to remove. On the non-generated corpora, TPR rises from 0.021 to 0.835 on XSum and from 0.251 to 0.879 on Wikipedia, but only from 0.017 to 0.039 on CC-News. Appendix F adds WBC, uncertainty, other operating points, target and reference sizes, LoRA, and DP-SGD; Appendix N tracks the signal over training.

Contributions of centering and weighting. Centering supplies the main improvement: mean TPR at 0.1% FPR rises from 0.112 to 0.418, while weighting the observed gain leaves it at 0.110 (Table 3). Once gains are centered, variance weighting with $\tau = 1$ adds 0.056 TPR. Weighting therefore helps only after centering has removed the domain baseline. These ablations support the two steps in Equation (5): subtract the expected gain, then account for its variability. Further ablations examine the choice of τ, alternative centering terms, and full-null transformations (Appendices G to I); calibrations learned with membership labels do not outperform the closed form (Appendix J).

Sensitivity to text and model access. At 256 tokens, RATIO+DIANA still more than doubles WBC's detection rate: 20.7% versus 9.7% mean TPR (Figure 4a). For $\tau \in \{0.25, 0.5, 1, 2, 4\}$, AUC stays within 0.005 of the best tested value on every corpus, supporting a fixed τ without pool estimation (panel b; per-corpus curves in Appendix I). Under restricted access, even the union of the models' top-5 tokens supplies useful moments: mean TPR is 30.3%, versus 7.6% for uncalibrated DIFF (panel c). This uses both models' probabilities on the support union; Appendix K gives the access protocol and per-corpus results.

5 RELATED WORK

Membership inference on language models has been studied in pre-training (Shi et al., 2023; Duan et al., 2024) and fine-tuning (Mireshghallah et al., 2022b). Early attacks threshold the loss (Yeom et al., 2018) or train shadow models (Shokri et al., 2017), and white-box attacks additionally use gradients or parameter perturbations (Nasr et al., 2019; Li et al., 2023). Likelihood-based attacks calibrate for sequence difficulty through reference models (Watson et al., 2021; Mireshghallah et al., 2022a) or neighborhood comparisons (Mattern et al., 2023), and shadow-model attacks use auxiliary data to estimate score distributions across training runs (Carlini et al., 2022; Ye et al., 2022; Hayes et al., 2025). Min-K% averages the lowest token log-probabilities (Shi et al., 2023), and Min-K%++ first standardizes them under the target's predictive distribution (Zhang et al., 2025). DC-PDD calibrates token probabilities with corpus token frequencies (Zhang et al., 2024), and ReCall and Con-ReCall compare likelihoods conditioned on known member or non-member prefixes (Xie et al., 2024; Wang et al., 2025). In pre-training, systematic evaluations find that such scores discriminate weakly and that text-only classifiers can match them when members and non-members differ in distribution (Duan et al., 2024; Das et al., 2025). For fine-tuned models, WBC (Chen et al., 2026) aggregates windowed signs of gains to tolerate sparse contamination, and EZ-MIA (Ilić et al., 2026) aggregates gains where the target's top prediction is incorrect. DIANA instead pursues the non-member comparison of shadow-model attacks with a token-level baseline estimated from the target's predictive distribution, without auxiliary data or additional training. Appendix C extends this discussion to memorization, other membership settings and defenses. We compare DIANA with these text-only, reference-free, white-box, and reference-based attacks on the same target sequences (Section 4.1), where RATIO+DIANA outperforms all of them in mean AUC and TPR at low FPR.

6 CONCLUSION

We introduced DIANA, a calibration for reference-based membership inference on fine-tuned language models that doesn't require the access to auxiliary data or target distribution. DIANA subtracts the target-weighted mean gain from each observed gain and divides by the regularized gain variance. As a drop-in replacement for the observed gain, DIANA strengthens DIFF, RATIO, WBC, and EZ-MIA. RATIO+DIANA raises mean TPR at 0.1% FPR from 0.323 for the strongest prior attack to 0.474 and transfers across five further model families and three non-generated corpora. *Limitations.* DIANA needs both models' next-token probabilities, although the top 50 recover most of its gain, and DP-SGD training defeats it along with every compared attack.

Ethics Statement

Membership inference supports privacy auditing but can also expose whether a person's data was used for training. Our experiments use publicly available corpora and models fine-tuned under controlled membership splits. We study aggregate attack performance rather than attributing sensitive records to individuals. Improved attacks should be used for authorized privacy assessments, with target records and inferred membership treated as sensitive information. The repository provides code for preparing data from the original sources.

Reproducibility Statement

Section 3 and appendix B specify the calibrated gain and its implementation, and Appendix A provides complete proofs with explicit assumptions. Appendix D documents data preparation, training, baseline implementations, and evaluation; Appendix E reports the full primary results and uncertainty. The DIANA repository includes data preparation, target training, feature collection, and attack evaluation with RATIO and WBC, together with links to sources for the remaining baselines. The code is available at `https://anonymous.4open.science/r/Diana/`.

AI Use Statement

Generative AI assisted with table construction, code preparation, and manuscript editing. The authors reviewed and verified all AI-assisted content and take responsibility for the final paper.

References

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

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. Cosmopedia, February 2024. URL `https://huggingface.co/datasets/HuggingFaceTB/cosmopedia`.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A suite for analyzing large language models across training and scaling. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/biderman23a.html`.

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

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

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://arxiv.org/abs/2202.07646`.

Hongyan Chang, Ali Shahin Shamsabadi, Kleomenis Katevas, Hamed Haddadi, and Reza Shokri. Context-aware membership inference attacks against pre-trained large language models. *arXiv preprint arXiv:2409.13745*, 2024.

Yuetian Chen, Yuntao Du, Kaiyuan Zhang, Ashish Kundu, Charles Fleming, Bruno Ribeiro, and Ninghui Li. Window-based membership inference attacks against fine-tuned large language models. *arXiv preprint arXiv:2601.02751*, 2026.

Christopher A. Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks, 2021.

Debeshee Das, Jie Zhang, and Florian Tramèr. Blind baselines beat membership inference attacks for foundation models, 2025. URL https://arxiv.org/abs/2406.16201.

Yuntao Du, Jiacheng Li, Yuetian Chen, Kaiyuan Zhang, Zhizhen Yuan, Hanshen Xiao, Bruno Ribeiro, and Ninghui Li. Cascading and Proxy Membership Inference Attacks. In *33rd Annual Network and Distributed System Security Symposium (NDSS)*, 2026.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*, 2024.

Qizhang Feng, Siva Rajesh Kasa, Santhosh Kumar Kasa, Hyokun Yun, Choon Hui Teo, and Sravan Babu Bodapati. Exposing privacy gaps: Membership inference attack on preference data for llm alignment, 2025. URL https://arxiv.org/abs/2407.06443.

Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. Mia-tuner: adapting large language models as pre-training text detector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 27295–27303, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024. URL https://arxiv.org/abs/2312.00752.

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. news-please: A generic news crawler and extractor. In *Proceedings of the 15th International Symposium of Information Science*, pp. 218–223, March 2017. doi: 10.5281/zenodo.4120316.

Jamie Hayes, Ilia Shumailov, Christopher A. Choquette-Choo, Matthew Jagielski, Georgios Kaissis, Milad Nasr, Meenatchi Sundaram Muthu Selva Annamalai, Niloofar Mireshghallah, Igor Shilov, Matthieu Meeus, Yves-Alexandre de Montjoye, Katherine Lee, Franziska Boenisch, Adam Dziedzic, and A. Feder Cooper. Exploring the limits of strong membership inference attacks on large language models. In *Advances in Neural Information Processing Systems*, 2025.

Yu He, Boheng Li, Liu Liu, Zhongjie Ba, Wei Dong, Yiming Li, Zhan Qin, Kui Ren, and Chun Chen. Towards label-only membership inference attack against pre-trained large language models, 2025. URL https://arxiv.org/abs/2502.18943.

Yuke Hu, Zheng Li, Zhihao Liu, Yang Zhang, Zhan Qin, Kui Ren, and Chun Chen. Membership inference attacks against vision-language models, 2025. URL https://arxiv.org/abs/2501.18624.

David Ilić, David Stanojević, and Kostadin Cvejoski. Powerful training-free membership inference against fine-tuned autoregressive language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14077–14093. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.640. URL https://aclanthology.org/2026.acl-long.640/.

Nikhil Kandpal, Krishna Pillutla, Alina Oprea, Peter Kairouz, Christopher A. Choquette-Choo, and Zheng Xu. User inference attacks on large language models, 2024. URL https://arxiv.org/abs/2310.09266.

Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. Copyright violations and large language models. *arXiv preprint arXiv:2310.13771*, 2023.

Klas Leino and Matt Fredrikson. Stolen memories: Leveraging model memorization for calibrated White-Box membership inference. In *29th USENIX security symposium (USENIX Security 20)*, pp. 1605–1622, 2020.

Hao Li, Zheng Li, Siyuan Wu, Chengrui Hu, Yutong Ye, Min Zhang, Dengguo Feng, and Yang Zhang. Seqmia: Sequential-metric based membership inference attack, 2024. URL https://arxiv.org/abs/2407.15098.

Marvin Li, Jason Wang, Jeffrey Wang, and Seth Neel. Mope: Model perturbation based privacy attacks on language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13647–13660, 2023.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. Not all tokens are what you need for pretraining. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 29029–29063. Curran Associates, Inc., 2024. doi: 10.52202/079017-0914. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/3322a9a72a1707de14badd5e552ff466-Paper-Conference.pdf.

Yiyong Liu, Zhengyu Zhao, Michael Backes, and Yang Zhang. Membership inference attacks by exploiting loss trajectory, 2022. URL https://arxiv.org/abs/2208.14933.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.

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

Matthieu Meeus, Shubham Jain, Marek Rei, and Yves-Alexandre de Montjoye. Did the neurons read your book? document-level membership inference for large language models. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 2369–2385, 2024.

Matthieu Meeus, Igor Shilov, Shubham Jain, Manuel Faysse, Marek Rei, and Yves-Alexandre de Montjoye. Sok: Membership inference attacks on llms are rushing nowhere (and how to fix it), 2025. URL https://arxiv.org/abs/2406.17975.

Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. Quantifying privacy risks of masked language models using membership inference attacks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8332–8347, 2022a.

Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David Evans, and Taylor Berg-Kirkpatrick. An empirical analysis of memorization in fine-tuned autoregressive language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 1816–1826, 2022b.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, et al. Olmoe: Open mixture-of-experts language models. In *International Conference on Learning Representations*, volume 2025, pp. 62061–62121, 2025.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://aclanthology.org/D18-1206/.

Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pp. 739–753. IEEE, 2019.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023.

Haritz Puerto, Martin Gubri, Sangdoo Yun, and Seong Joon Oh. Scaling up membership inference: When and how attacks succeed on large language models, 2025. URL https://arxiv.org/abs/2411.00154.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. *arXiv preprint arXiv:2406.05946*, 2024.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pp. 5558–5567. PMLR, 2019.

Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society, 2019.

Ali Satvaty, Suzan Verberne, and Fatih Turkmen. Undesirable memorization in large language models: A survey, 2025. URL https://arxiv.org/abs/2410.02650.

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

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022.

Cheng Wang, Yiwei Wang, Bryan Hooi, Yujun Cai, Nanyun Peng, and Kai-Wei Chang. Con-recall: Detecting pre-training data in llms via contrastive decoding. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 1013–1026, 2025.

Lauren Watson, Chuan Guo, Graham Cormode, and Alex Sablayrolles. On the importance of difficulty calibration in membership inference attacks. *arXiv preprint arXiv:2111.08440*, 2021.

Rui Wen, Zheng Li, Michael Backes, and Yang Zhang. Membership inference attacks against in-context learning, 2024. URL https://arxiv.org/abs/2409.01380.

Wikimedia Foundation. Wikimedia downloads, 2023. URL https://dumps.wikimedia.org. English Wikipedia snapshot 20231101.en; distributed at https://huggingface.co/datasets/wikimedia/wikipedia.

Roy Xie, Junlin Wang, Ruomin Huang, Minxing Zhang, Rong Ge, Jian Pei, Neil Zhenqiang Gong, and Bhuwan Dhingra. Recall: Membership inference via relative conditional log-likelihoods. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8671–8689, 2024.

Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC conference on computer and communications security*, pp. 3093–3106, 2022.

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

Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500*, 2021.

Shenglai Zeng, Yaxin Li, Jie Ren, Yiding Liu, Han Xu, Pengfei He, Yue Xing, Shuaiqiang Wang, Jiliang Tang, and Dawei Yin. Exploring memorization in fine-tuned language models. *arXiv preprint arXiv:2310.06714*, 2023.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. Min-k%++: Improved baseline for detecting pre-training data from large language models, 2025.

Ruisi Zhang, Seira Hidano, and Farinaz Koushanfar. Text revealer: Private text reconstruction via model inversion attacks against transformers. *arXiv preprint arXiv:2209.10505*, 2022a.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: Open pre-trained transformer language models, 2022b. URL https://arxiv.org/abs/2205.01068.

Weichao Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. Pretraining data detection for large language models: A divergence-based calibration method. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5263–5274, 2024.

A Proofs of the Theoretical Results

We use the same gains G_j, s_j, and $\tilde{s}_j$ and moments μ_j and σ_j^2 as in Equation (5). Each proof specifies which prediction is used to compute them and what remains random. The non-member and member proofs first establish the mean of the centered gain, then pass to the calibrated gain. For members, this step uses a general bound on weighted means (Lemma 1). The concavity proof in Appendix A.3 reuses the member mean derived in Appendix A.2.

A.1 Proposition 1: Conditional Null Centering

Proposition 1 (Exact conditional null centering)**.** *For a target sequence* $\mathbf{x}$ *not used in training* $\mathcal{M}^{\mathrm{T}}$*, under the assumption* $p_j^- = p_j^{\mathcal{D}}$*, every scored position* $j \in \{2, \ldots, n\}$ *satisfy*

$$\mathbb{E}[\tilde{s}_j] = 0, \tag{6}$$

where $\tilde{s}_j$ *is as defined in Equation* (5)*.*

Proof of Proposition 1. **Step 1: Fix the distributions and the centering constant.** Fix a prefix $x_{<j}$ and the non-member case $M = 0$. Hold the domain law $p_j^{\mathcal{D}}$, the reference law p_j^{R}, and the idealized non-member prediction p_j^- fixed. Evaluate the gains and moments in Equation (5) using p_j^-. Full support means that each token has positive probability; on the finite vocabulary, this makes every gain finite:

$$G_j(v) = \log \frac{p_j^-(v)}{p_j^{\mathrm{R}}(v)}, \qquad v \in \mathcal{V}, \tag{9}$$

and

$$\mu_j = \sum_v p_j^-(v) G_j(v), \qquad s_j = G_j(X_j) - \mu_j. \tag{10}$$

Only the next token $X_j \sim p_j^{\mathcal{D}}$ is random. The centering constant μ_j and variance σ_j^2 are computed from the fixed prediction, so neither changes with the realized token. All expectations in this proof are conditional on the fixed prefix and non-membership.

Step 2: Average the observed gain and subtract the center. Expanding the definition gives

$$\begin{aligned}\mathbb{E}[s_j] &= \sum_v p_j^{\mathcal{D}}(v)\,[G_j(v) - \mu_j] \\ &= \sum_v p_j^{\mathcal{D}}(v) G_j(v) - \mu_j \sum_v p_j^{\mathcal{D}}(v) \\ &= \sum_v p_j^{\mathcal{D}}(v) G_j(v) - \sum_v p_j^{-}(v) G_j(v).\end{aligned}$$

Combining the sums gives the residual when the two distributions differ:

$$\mathbb{E}[s_j] = \sum_v \big(p_j^{\mathcal{D}}(v) - p_j^{-}(v)\big) G_j(v). \tag{11}$$

This establishes Equation (11) without assuming that the non-member prediction matches the domain law.

Step 3: Apply the domain-matching assumption. When $p_j^{-} = p_j^{\mathcal{D}}$, every difference in Equation (11) is zero. Hence $\mathbb{E}[s_j] = 0$: the expected observed gain equals the mean removed by centering. This cancellation holds for any full-support reference distribution.

Step 4: Divide by the fixed positive denominator. The target-weighted variance here is $\sigma_j^2 = \mathrm{Var}_{W \sim p_j^{-}}[G_j(W)]$. It is finite by Step 1 and nonnegative by the definition of variance. Since $\tau > 0$, the denominator is positive and can be taken outside the expectation:

$$\mathbb{E}[\tilde{s}_j] = \mathbb{E}\left[\frac{s_j}{\tau + \sigma_j^2}\right] = \frac{\mathbb{E}[s_j]}{\tau + \sigma_j^2} = 0.$$

If $p_j^{-} \neq p_j^{\mathcal{D}}$, the same calculation divides the residual in Equation (11) by $\tau + \sigma_j^2$. Therefore, for a target sequence not used in training and under $p_j^{-} = p_j^{\mathcal{D}}$, every $\tau > 0$ and scored position j satisfy $\mathbb{E}[\tilde{s}_j] = 0$, which proves Proposition 1. □

A.2 Theorem 1: Positive Conditional Member Mean

Theorem 1 (Positive conditional member mean)**.** *For a target sequence* $\mathbf{x}$ *included in training* $\mathcal{M}^{\mathrm{T}}$*,* $\exists \tau > 0$ *such that under the member next-token distribution* p_j^{+} *in Equation* (7)*, each scored position* $j \in \{2, \ldots, n\}$ *satisfies*

$$\mathbb{E}[\tilde{s}_j] > 0, \tag{8}$$

where $\tilde{s}_j$ *is as defined in Equation* (5)*.*

Roadmap. The proof has two parts. Part 1 (Steps 1–4) computes the centered gain for each possible memorized token and shows that its average is positive. Part 2 (Step 5) passes to the calibrated gain. This step needs a separate argument, because the variance, and hence the weight $1/(\tau + \sigma_j^2)$, depends on which token is memorized, and unequal positive weights can change the sign of an average that contains both positive and negative values. Lemma 1 bounds how unequal the weights may be.

Lemma 1 (Bounded weights preserve a positive mean)**.** *Let* S *and* W *be random variables with finitely many values and* $0 < w_{\min} \leq W \leq w_{\max}$*. Let* $m \coloneqq \mathbb{E}[S] > 0$ *be the mean of* S*, and let* $N \coloneqq \mathbb{E}[\max\{-S, 0\}]$ *be the average size of its negative values, with* $N = 0$ *if* S *is never negative. If*

$$\frac{w_{\max}}{w_{\min}} < 1 + \frac{m}{N},$$

where the right side is read as $+\infty$ *when* $N = 0$*, then* $\mathbb{E}[WS] > 0$*.*

Intuition. Reweighting can lower a positive mean only by giving the negative values of S more weight than its positive values. On average, the positive values contribute $m + N$ and the negative

values contribute N, so the ratio m/N measures how strongly the positive values dominate. The lemma says that the mean stays positive as long as the largest weight is less than $1 + m/N$ times the smallest weight.

Proof. Split $S = S^+ - S^-$ into its positive part $S^+ = \max\{S, 0\}$ and negative part $S^- = \max\{-S, 0\}$, so that $\mathbb{E}[S^-] = N$ and $\mathbb{E}[S^+] = \mathbb{E}[S] + \mathbb{E}[S^-] = m + N$. Both parts are non-negative, so the worst case gives every positive value the smallest weight and every negative value the largest weight:

$$\mathbb{E}[WS] = \mathbb{E}[WS^+] - \mathbb{E}[WS^-] \geq w_{\min}(m + N) - w_{\max}N.$$

If $N = 0$, the right side equals $w_{\min}m > 0$. Otherwise, dividing by $w_{\min}N > 0$ shows that the right side is positive exactly when $w_{\max}/w_{\min} < 1 + m/N$. □

Proof of Theorem 1. **Step 1: Setup.** Fix a prefix $x_{<j}$, full-support domain and reference distributions on a finite vocabulary with at least two tokens, and $0 < \alpha_j < 1$ in Equation (7). For a member, the observed token x_j is also a memorized token. As in Proposition 1, we treat it as random: $Z \sim p_j^{\mathcal{D}}$, with z denoting one of its values. Memorizing z gives the prediction

$$p_{j,z}^+(v) = (1 - \alpha_j)p_j^{\mathcal{D}}(v) + \alpha_j \mathbf{1}_{\{v=z\}}, \qquad v \in \mathcal{V}.$$

All quantities in Equation (5) are computed from this prediction and therefore depend on z; we write $s_j(z)$, $\sigma_j^2(z)$, and $\tilde{s}_j(z) = s_j(z)/(\tau + \sigma_j^2(z))$. Expectations average over the memorized token, $\mathbb{E}[f(Z)] = \sum_z p_j^{\mathcal{D}}(z)f(z)$, with the domain law, the reference law, and α_j held fixed. Within this proof we abbreviate

$$b_j(v) \coloneqq \log \frac{p_j^{\mathcal{D}}(v)}{p_j^{\mathrm{R}}(v)}, \qquad \bar{b}_j \coloneqq \sum_v p_j^{\mathcal{D}}(v)b_j(v), \qquad c_j(z) \coloneqq \log\left(1 + \frac{\alpha_j}{(1 - \alpha_j)p_j^{\mathcal{D}}(z)}\right) > 0.$$

Here b_j is the domain-to-reference log-ratio and $c_j(z)$ is the extra log-probability that memorization adds to token z.

Part 1: the centered gain.

Step 2: The gain function. For $v \neq z$, the member prediction is $(1 - \alpha_j)p_j^{\mathcal{D}}(v)$. For $v = z$, it is $(1 - \alpha_j)p_j^{\mathcal{D}}(z) + \alpha_j = (1 - \alpha_j)p_j^{\mathcal{D}}(z)\, e^{c_j(z)}$. Dividing by $p_j^{\mathrm{R}}(v)$ and taking logarithms gives

$$b_j(v) + \log(1 - \alpha_j) + c_j(z)\mathbf{1}_{\{v=z\}} = G_j(v), \qquad v \in \mathcal{V},$$

which is the gain $G_j(v)$.

Step 3: The centered gain. Under $p_{j,z}^+$, the target-weighted mean is $\mu_j = (1-\alpha_j)\sum_v p_j^{\mathcal{D}}(v)G_j(v) + \alpha_j G_j(z)$, so

$$s_j(z) = G_j(z) - \mu_j = (1 - \alpha_j)\Big[G_j(z) - \sum_v p_j^{\mathcal{D}}(v)G_j(v)\Big].$$

By Step 2, $G_j(z) = b_j(z) + \log(1-\alpha_j) + c_j(z)$ and $\sum_v p_j^{\mathcal{D}}(v)G_j(v) = \bar{b}_j + \log(1-\alpha_j) + p_j^{\mathcal{D}}(z)c_j(z)$. Subtracting cancels $\log(1 - \alpha_j)$:

$$s_j(z) = \underbrace{(1 - \alpha_j)\big[b_j(z) - \bar{b}_j\big]}_{\text{baseline deviation}} + \underbrace{(1 - \alpha_j)\big(1 - p_j^{\mathcal{D}}(z)\big)c_j(z)}_{\text{memorization contribution}}. \tag{12}$$

The baseline deviation can have either sign; the memorization contribution is strictly positive.

Step 4: The mean centered gain. Because $Z \sim p_j^{\mathcal{D}}$, $\mathbb{E}[b_j(Z)] = \bar{b}_j$, so the baseline deviation averages to zero and

$$m_j \coloneqq \mathbb{E}[s_j(Z)] = (1 - \alpha_j)\sum_z p_j^{\mathcal{D}}(z)\big(1 - p_j^{\mathcal{D}}(z)\big)\log\left(1 + \frac{\alpha_j}{(1 - \alpha_j)p_j^{\mathcal{D}}(z)}\right). \tag{13}$$

Full support and at least two tokens give $0 < p_j^{\mathcal{D}}(z) < 1$, so every summand is positive and $m_j > 0$. Individual centered gains can still be negative.

Part 2: the calibrated gain.

Step 5: The mean calibrated gain. We now show that dividing by $\tau + \sigma_j^2$ keeps the mean positive once τ is large enough. We proceed in five short steps.

- *What changes.* The calibrated gain is the centered gain multiplied by a weight, $\tilde{s}_j(z) = W(z)s_j(z)$ with $W(z) \coloneqq 1/(\tau + \sigma_j^2(z))$. For a non-member, the variance is computed from a prediction that does not depend on the observed token, so the weight is a constant and the mean keeps its sign (Appendix A.1). For a member, the prediction $p_{j,z}^+$ depends on which token z is memorized, so the variance $\sigma_j^2(z)$ and the weight $W(z)$ vary with z. If negative centered gains happened to receive larger weights than positive ones, the weighted mean could become negative. Lemma 1 bounds how unequal the weights can be before this happens, with $S = s_j(Z)$ and $W = W(Z)$.

- *The quantities in the lemma.* The mean is $m_j = \mathbb{E}[s_j(Z)] > 0$ from Step 4. The negative part is $N_j \coloneqq \mathbb{E}[\max\{-s_j(Z), 0\}]$, the average size of the negative centered gains; these arise from the baseline deviation in Equation (12) when the memorized token's domain-to-reference log-ratio is below its domain average. Let $\sigma^2_{j,\min}$ and $\sigma^2_{j,\max}$ be the smallest and largest variances over the finitely many possible memorized tokens. All four quantities are determined by the domain law, the reference law, and α_j; none depends on τ.

- *How τ limits the weight ratio.* For every z, the weight lies between $w_{\min} \coloneqq 1/(\tau + \sigma^2_{j,\max})$ and $w_{\max} \coloneqq 1/(\tau + \sigma^2_{j,\min})$. Their ratio is

$$\frac{w_{\max}}{w_{\min}} = \frac{\tau + \sigma^2_{j,\max}}{\tau + \sigma^2_{j,\min}} = 1 + \frac{\sigma^2_{j,\max} - \sigma^2_{j,\min}}{\tau + \sigma^2_{j,\min}},$$

  which decreases toward one as τ grows: a larger τ makes the weights more nearly equal.

- *Applying the lemma.* The lemma's condition $w_{\max}/w_{\min} < 1 + m_j/N_j$ becomes $(\sigma^2_{j,\max} - \sigma^2_{j,\min})/(\tau + \sigma^2_{j,\min}) < m_j/N_j$. Multiplying both sides by $N_j(\tau + \sigma^2_{j,\min}) \geq 0$ gives

$$N_j\big(\sigma^2_{j,\max} - \sigma^2_{j,\min}\big) < m_j\big(\tau + \sigma^2_{j,\min}\big), \tag{14}$$

  which also covers $N_j = 0$. Whenever Equation (14) holds, the lemma gives $\mathbb{E}[\tilde{s}_j(Z)] > 0$.

- *Solving for τ.* The left side of Equation (14) does not depend on τ, and the right side grows linearly in τ with slope $m_j > 0$. Rearranging, the condition holds for every $\tau > \tau_j^*$, where

$$\tau_j^* \coloneqq \max\left\{0,\ \frac{N_j}{m_j}\big(\sigma^2_{j,\max} - \sigma^2_{j,\min}\big) - \sigma^2_{j,\min}\right\}.$$

  This threshold is finite because $m_j > 0$ and all other quantities are finite.

Therefore, for a target sequence included in training and under the single-spike model, every scored position j has a finite threshold τ_j^* such that every $\tau > \tau_j^*$ yields $\mathbb{E}[\tilde{s}_j] > 0$. □

Reading the threshold. The threshold τ_j^* grows with two factors. The first, N_j/m_j, compares the negative centered gains with the positive mean: the more negative mass there is, the more equal the weights must be. The second, $\sigma^2_{j,\max} - \sigma^2_{j,\min}$, is the spread of the variance across possible memorized tokens: it is the only source of unequal weights. If the variance does not change with the memorized token, the spread is zero, $\tau_j^* = 0$, and every $\tau > 0$ preserves the positive mean. Taking τ above $\max_j \tau_j^*$ gives one value for all scored positions. Because τ_j^* depends on the domain law, the reference law, and the memorization strength, none of which the attacker observes, the theorem shows that a suitable τ exists but does not fix its value. We use $\tau = 1$, supported by the τ sweep (Section 4.2 and appendix I) and by the positive calibrated member mean at every checkpoint of the training ladder (Appendix N).

A.3 Proposition 2: Concavity in Memorization Strength

Proposition 2 (Concavity in memorization strength)**.** *Under the single-spike assumptions of Theorem 1, the expected centered member gain $\mathbb{E}[s_j]$ is strictly concave in α_j (memorization strength).*

Proof of Proposition 2. We use the setting of Appendix A.2, where Steps 1–4 derive the centered member mean in Equation (13).

Step 1: Write one memorization contribution in terms of token probability. Fix one possible memorized token z. Its probability after memorization is $u = p_j^{\mathcal{D}}(z) + (1 - p_j^{\mathcal{D}}(z))\alpha_j$, so $1 - u = (1 - \alpha_j)(1 - p_j^{\mathcal{D}}(z))$. Substitution gives

$$(1 - \alpha_j)(1 - p_j^{\mathcal{D}}(z)) \log\left(1 + \frac{\alpha_j}{(1 - \alpha_j)p_j^{\mathcal{D}}(z)}\right) = (1 - u) \log \frac{u(1 - p_j^{\mathcal{D}}(z))}{p_j^{\mathcal{D}}(z)(1 - u)}.$$

This is one summand of Equation (13) before multiplication by the fixed domain probability $p_j^{\mathcal{D}}(z)$.

Step 2: Differentiate twice. Holding $p_j^{\mathcal{D}}(z)$ fixed, the first derivative of the preceding expression is

$$\frac{d}{du}\left[(1 - u) \log \frac{u(1 - p_j^{\mathcal{D}}(z))}{p_j^{\mathcal{D}}(z)(1 - u)}\right] = -\log \frac{u(1 - p_j^{\mathcal{D}}(z))}{p_j^{\mathcal{D}}(z)(1 - u)} + \frac{1}{u}.$$

Differentiating once more gives

$$\frac{d^2}{du^2}\left[(1 - u) \log \frac{u(1 - p_j^{\mathcal{D}}(z))}{p_j^{\mathcal{D}}(z)(1 - u)}\right] = -\frac{1}{u(1 - u)} - \frac{1}{u^2} = -\frac{1}{u^2(1 - u)} < 0. \tag{15}$$

The probability u is affine in α_j, with derivative $1 - p_j^{\mathcal{D}}(z)$. The chain rule therefore multiplies each second derivative by $(1 - p_j^{\mathcal{D}}(z))^2$. Averaging with the fixed probabilities $p_j^{\mathcal{D}}(z)$ in Equation (13) yields

$$\frac{d^2}{d\alpha_j^2}\mathbb{E}[s_j] = -\sum_z \frac{p_j^{\mathcal{D}}(z)(1 - p_j^{\mathcal{D}}(z))}{(1 - \alpha_j)[p_j^{\mathcal{D}}(z) + (1 - p_j^{\mathcal{D}}(z))\alpha_j]^2} < 0.$$

Therefore, under the single-spike assumptions, the expected centered member gain $\mathbb{E}[s_j]$ is strictly concave in the memorization strength α_j, which proves Proposition 2.

□

Empirically, the centered and calibrated member means both rise and then fall over training; at $\tau = 1$, the calibrated mean remains positive at every checkpoint (Appendix N).

B ALGORITHMIC SPECIFICATION

Algorithm 1 computes the calibrated gains $\tilde{s}_j$ and passes them to the host attack's aggregation rule $\mathcal{A}$.

`GainMoments` (Lines 1–8) needs only one forward pass of each model over $\mathbf{x}$, because both models are causal. At each prefix, it forms the gain vector over the vocabulary and its target-weighted moments: inner products sum over the vocabulary, $G_j^{\odot 2}$ squares each entry, and the maximum in Line 6 removes negative roundoff. The hat marks a variance computed from the trained target. Lines 11–12 center each observed gain and divide it by $\tau + \hat{\sigma}_j^2$, as in Equation (5); we use $\tau = 1$ throughout (Appendix I). Line 13 keeps the host rule; for example, RATIO+DIANA replaces G_j^{obs} with $\tilde{s}_j$ in Equation (4), and the resulting score is thresholded to predict membership. With top-k probabilities (Appendix K), Lines 5–6 use both models' renormalized probabilities on the union of their top-k tokens, and Line 7 stays exact.

Algorithm 1: Attack scoring with DIANA

input : models $\mathcal{M}^{\mathrm{T}}, \mathcal{M}^{\mathrm{R}}$; target sequence $\mathbf{x} = (x_1, \ldots, x_n)$; aggregation rule $\mathcal{A}$; hyperparameter $\tau > 0$
output : attack score for $\mathbf{x}$

1 **function** `GainMoments(`$\mathbf{x}$`)`**:**
2 **for** $j \leftarrow 2$ **to** n **do**
3 $p_j^{\mathrm{T}} \leftarrow \mathcal{M}^{\mathrm{T}}(\cdot \mid x_{<j})$; $p_j^{\mathrm{R}} \leftarrow \mathcal{M}^{\mathrm{R}}(\cdot \mid x_{<j})$
4 $G_j \leftarrow \log p_j^{\mathrm{T}} - \log p_j^{\mathrm{R}}$
5 $\mu_j \leftarrow \langle p_j^{\mathrm{T}}, G_j \rangle$
6 $\hat{\sigma}_j^2 \leftarrow \max\{\langle p_j^{\mathrm{T}}, G_j^{\odot 2} \rangle - \mu_j^2, 0\}$
7 $G_j^{\text{obs}} \leftarrow G_j(x_j)$
8 **return** $(G^{\text{obs}}, \mu, \hat{\sigma}^2)$
9 $(G^{\text{obs}}, \mu, \hat{\sigma}^2) \leftarrow$ `GainMoments(`$\mathbf{x}$`)`
10 **for** $j \leftarrow 2$ **to** n **do**
11 $s_j \leftarrow G_j^{\text{obs}} - \mu_j$
12 $\tilde{s}_j \leftarrow s_j / (\tau + \hat{\sigma}_j^2)$
13 **return** $\mathcal{A}(\tilde{s}_2, \ldots, \tilde{s}_n)$

C EXPANDED RELATED WORK

Attack access and difficulty calibration. Early membership inference attacks use shadow models (Shokri et al., 2017; Salem et al., 2019) or a loss threshold (Yeom et al., 2018). Subsequent work analyzes Bayes-optimal membership tests (Sablayrolles et al., 2019), develops white-box attacks using model internals (Nasr et al., 2019; Leino & Fredrikson, 2020; Li et al., 2023), and studies label-only access (Choquette-Choo et al., 2021; He et al., 2025). Difficulty calibration adjusts for the fact that target sequences can receive favorable scores without being members. Reference-model comparisons (Watson et al., 2021; Mireshghallah et al., 2022a), likelihood-ratio tests built from shadow-model scores (Carlini et al., 2022; Ye et al., 2022), and cascading or proxy attacks (Du et al., 2026) pursue this objective with different modeling and access requirements. Shadow-model score distributions describe how a fixed target sequence is scored across training runs; estimating them at LLM scale requires auxiliary data and repeated model training (Hayes et al., 2025). DIANA adopts the same counterfactual question: what gain could arise without training on the target sequence? It answers at each prefix by computing gain moments across possible continuations under the queried target. The target supplies a proxy for the domain law $p_j^{\mathcal{D}}$; for idealized mean predictions satisfying $p_j^- = p_j^{\mathcal{D}}$, Proposition 1 establishes exact null centering. This provides a token-level non-member baseline without training shadow models. The moments describe variation across continuations of a fixed model, whereas shadow-model score distributions describe variation across training runs.

Memorization and text recovery. Extraction attacks demonstrate that language models can reproduce training text (Carlini et al., 2021; Nasr et al., 2023), while memorization studies examine its prevalence and training dynamics (Carlini et al., 2023; Tirumala et al., 2022; Satvaty et al., 2025). Related work investigates private-text reconstruction (Zhang et al., 2022a) and copyright leakage (Karamolegkou et al., 2023). Recovering memorized text and determining whether a supplied target sequence was used for training probe different aspects of this behavior. Systematic evaluations report limited membership discrimination in pre-training settings (Duan et al., 2024; Das et al., 2025;

Meeus et al., 2025); studies of fine-tuned models document substantial memorization and membership leakage (Mireshghallah et al., 2022b; Zeng et al., 2023).

Language-model membership scores. Min-K% averages the lowest token log-probabilities (Shi et al., 2023), and Min-K%++ applies lower-tail aggregation after standardizing log-probabilities under the target distribution (Zhang et al., 2025). DC-PDD calibrates probabilities using corpus token frequencies (Zhang et al., 2024); neighborhood attacks compare a target sequence with perturbed texts (Mattern et al., 2023). ReCall and Con-ReCall use conditional likelihoods and contrastive scoring (Xie et al., 2024; Wang et al., 2025), and other work develops context-aware attacks (Chang et al., 2024) or adapts language models as membership detectors (Fu et al., 2025). Min-K%++ shares the use of predictive moments with DIANA: it standardizes a single model's log-probabilities, whereas DIANA centers target-to-reference gains and weights them by their regularized variance. Appendix D gives the access and implementation details of our evaluated baselines.

Gain aggregation after fine-tuning. Reference-based attacks score changes in a target sequence's likelihood relative to a reference model (Carlini et al., 2021; Watson et al., 2021). WBC (Chen et al., 2026) decomposes observed gains into membership signal, sparse contamination, and dense noise, motivating a windowed sign-test ensemble. EZ-MIA (Ilić et al., 2026) selects positions where the target's most probable token is incorrect and compares positive and negative gains within those positions. DIANA uses WBC's signal model to motivate estimating the gain expected from ordinary domain continuations. Under the idealized domain-matching model, the calibrated gain has zero conditional mean for non-members (Proposition 1). The calibrated gain can be passed to the existing aggregation rules, as evaluated in Table 1 and appendix E.

Other inference settings. Membership inference also considers document-level and user-level evidence (Meeus et al., 2024; Kandpal et al., 2024; Puerto et al., 2025), loss trajectories and sequential metrics (Liu et al., 2022; Li et al., 2024), examples supplied through in-context learning (Wen et al., 2024), preference data used for alignment (Feng et al., 2025), and vision-language models (Hu et al., 2025). These settings vary the membership unit, the training procedure, or the observations available to the attacker. Our experiments concern membership of a target sequence in a fine-tuning corpus.

Token selection and defenses. Unequal token contributions also motivate Rho-1's selection of pre-training tokens using excess loss against a reference model (Lin et al., 2024) and analyses of how safety alignment affects a small set of response positions (Qi et al., 2024). DP-SGD limits training-data influence through clipped, noisy gradient updates (Abadi et al., 2016; Yu et al., 2021); low-rank adaptation learns low-rank parameter updates. Appendix F evaluates membership inference under both training procedures. Unlearning benchmarks study the removal of specified knowledge after training (Maini et al., 2024).

D EVALUATION PROTOCOL AND BASELINES

Data and target models. The primary experiment uses six Cosmopedia corpora: Stanford, Khan Academy, Stories, AutoMathText, Web Samples v2, and WikiHow. Members and non-members are sampled from the same corpus and separated before fine-tuning. The primary splits contain 20,000 target sequences per class, except Khan Academy with 10,000 per class; the latter has only 21,688 documents satisfying the 512-token eligibility criterion. XSum, CC-News, and Wikipedia use 10,000 target sequences per class. The target model and its reference share a tokenizer within each comparison.

Corpus provenance. The non-generated corpora are the documents from `EdinburghNLP/xsum` (Narayan et al., 2018), articles from `vblagoje/cc_news` (Hamborg et al., 2017), and the `20231101.en` snapshot of `wikimedia/wikipedia` (Wikimedia Foundation, 2023). Our synthetic corpora come from the original `HuggingFaceTB/cosmopedia` release (Ben Allal et al., 2024).

Training. The main OLMOE targets use one epoch and learning rate 5×10^{-6}. Llama-3.1-8B uses the same schedule; Qwen2.5-1.5B uses three epochs at 10^{-5}. The PYTHIA 160M, 1B, and 6.9B targets use three epochs at 10^{-5}; the 410M, 1.4B, and 2.8B targets use the one-epoch settings below. Transfer, target-size, reference-size, full fine-tuning, and utility comparisons reuse the same

checkpoint for each target. The attack-evaluation ladder uses OLMoE at 2×10^{-6} for ten epochs, with three seeds and one checkpoint per 625 steps. The centered member-gain diagnostic in Figure 2(a) extends this recipe to twenty epochs and evaluates 800 members per checkpoint (Appendix N). LoRA and DP-SGD are separate training comparisons described in Appendix F.

Checkpoint selection. The XSum, CC-News, and Wikipedia targets use the checkpoints with the lowest token-weighted NLL on 2,000 additional validation documents per corpus, disjoint from both attack classes after text and token-prefix deduplication. Each is selected at epoch 1 and learning rate 5×10^{-6} from its saved trajectory.

Additional model recipes. Pythia-410M, Pythia-1.4B, Pythia-2.8B, and OPT-1.3B use one epoch at learning rates 2×10^{-5}, 6×10^{-6}, 5×10^{-6}, and 10^{-5}, respectively. Mamba-1.4B uses the first epoch of a ten-epoch trajectory at 3×10^{-5}. All use AdamW, effective batch size 32, weight decay 0.1, 100 warmup steps, linear decay, and seed 42; the scheduler horizon is one epoch for the dense targets and ten for Mamba. The dense targets use FP32 master weights with BF16 autocast. Validation uses 2,000 separate documents.

Reference-free baselines. Loss (Yeom et al., 2018) uses the target's sequence likelihood. ZLIB and Lowercase (Carlini et al., 2021) normalize by compressed text length and by likelihood after lowercasing, respectively. Equal token length does not make compressed byte length constant, so ZLIB remains distinct from Loss. Min-K% (Shi et al., 2023) averages the lowest 20% of token log-probabilities; Min-K%++ (Zhang et al., 2025) applies the corresponding lower-tail aggregation after standardizing target log-probabilities by their target-distribution moments. DC-PDD (Zhang et al., 2024) combines target probabilities with C4 token frequencies, using cap 0.01 and the first occurrence of each token type among the 511 predicted positions. Neighborhood (Mattern et al., 2023) compares the target sequence's likelihood with the mean likelihood of 50 single-token substitutions proposed by RoBERTa-base with embedding dropout 0.7. Substitutions lie within the masked model's context, and each neighbor retains the rest of the document. ReCall (Xie et al., 2024) conditions on four known non-member examples of at most 256 tokens each. Con-ReCall (Wang et al., 2025) contrasts seven such examples from each class with $\gamma = 1$; it additionally requires known members (Seq.+M in the tables). Both use five stratified folds: prefixes are drawn outside the scored fold, so every target sequence is evaluated without appearing in its own conditioning context. We score only target-sequence tokens 2–512, excluding prefix tokens from the likelihood. The text-only Bag-of-Words control (Das et al., 2025) uses a random forest trained on word-frequency features from the original documents and membership labels, with out-of-fold predictions from five stratified folds over the complete primary split. Its forest uses 100 trees, maximum depth 2, minimum leaf size 5, and a minimum document frequency of 0.05. Its AUC is 0.498–0.503 across the six primary splits, detecting no substantial separation from word frequencies.

White-box baselines. MoPe (Li et al., 2023) and GradNorm (Nasr et al., 2019) use target parameters and gradients, respectively, and are evaluated on the complete primary splits, as are all other baselines. MoPe averages ten weight perturbations with noise scale 0.002, selected using Stanford membership labels by maximizing its AUC over scales from 0.0005 to 0.01. GradNorm scores the negated norm of the loss gradient over all parameters.

Reference-based baselines. Diff and Ratio (Carlini et al., 2021; Watson et al., 2021) aggregate target-to-reference gains; WBC (Chen et al., 2026) uses windowed sign comparisons, with window lengths $\{2, 3, 4, 6, 9, 13, 18, 25, 32, 40\}$ in our implementation. Diana replaces their token gains with $\tilde{s}_j$ while retaining the host aggregation.

EZ-MIA (Ilić et al., 2026) uses the same pre-trained reference as Diff, Ratio, and WBC. It divides the sum of positive gains by the absolute sum of negative gains over positions where the target's most probable token differs from the observed token. Following the released implementation, we skip the first predicted position, require at least two remaining error positions, add 10^{-12} to a positive denominator, and return zero if the negative-gain sum is zero. None of the 220,000 target sequences triggers either fallback. EZ-MIA+Diana substitutes $\tilde{s}_j$ while retaining the error-position mask and aggregation; it also has no fallback cases. Both variants share fresh target and reference forward passes on the complete primary splits, with the same fixed $\tau = 1$. Top-1 access therefore

includes observed-token probabilities from both models and the target's most probable token; DIANA additionally uses both full-vocabulary distributions.

Scoring and uncertainty. All DIANA evaluations use $\tau = 1$, including the primary and transfer comparisons, training ladder, and calibrated baselines. The exceptions are the τ sweep and the ablation that sets $\tau = 0$. Appendix I shows robustness across positive values. Scoring requires no membership labels or statistics of the target-sequence pool. The full-split baseline evaluations use 500 sequence-bootstrap resamples with seed 0. Appendix performance tables report estimates with smaller $\pm$ bootstrap standard deviations. These intervals measure uncertainty from sampling target sequences for fixed target models; the three-seed ladder separately measures training variability. TPR is the largest empirical ROC value whose FPR is at most the stated budget. With $10{,}000$ and $20{,}000$ non-members, a 0.1% FPR budget allows at most 10 and 20 false positives, respectively.

Features and computation. DIANA and the gain-based baselines reuse target and reference forward passes on each target sequence. At each position, extraction records the observed-token log-probabilities, gain, mean gain, and second gain moment; the variance is the second moment minus the squared mean. The vocabulary reductions cost $O(V)$ per position and scoring these stored quantities is CPU-only. Lowercase, Min-K%++, conditioning-based attacks and the white-box attacks require their respective additional passes or features.

In attack-performance tables, the best and second best entries are bold and underlined, respectively.

E COMPLETE PRIMARY RESULTS AND SENSITIVITY DETAIL

The following tables report all six primary corpora, including Khan Academy, Stories, and Web Samples v2 beyond the three shown in Table 1. They give AUC and TPR at 10%, 1%, and 0.1% FPR. Each cell includes its recorded bootstrap standard deviation. The best result is bold and the second best is underlined. Target-sequence counts and access requirements are specified in Appendix D. RATIO+DIANA has the highest estimate at every operating point on each corpus. Absolute improvements need not increase monotonically as the FPR decreases; the operating points measure different portions of the ROC curve.

Prefix length and τ. The prefix study in Figure 4a scores the first $n \in \{16, 32, 64, 128, 256\}$ predicted positions of the same target sequences, with full scoring using all 511 positions in a 512-token record. All six saved evaluations retain their complete target-sequence counts at every displayed prefix length. The sweep in Appendix I varies τ from 0 to 16. The prefix study fixes $\tau = 1$ at every length and retains the complete target-sequence cohort.

Table 4: **OLMoE on Stanford.** Complete results with bootstrap standard deviations.

Type	Method	Access	AUC	TPR at FPR		
				10%	1%	**0.1%**
Text-only	Bag-of-Words	Text	.501 ± .003	.101 ± .003	.011 ± .001	.001 ± .000
Reference-free	Loss	Seq.	.674 ± .003	.270 ± .004	.070 ± .003	.012 ± .002
	ZLIB	Seq.	.608 ± .003	.190 ± .004	.030 ± .002	.006 ± .001
	Lowercase	Seq.	.680 ± .002	.281 ± .005	.054 ± .003	.013 ± .002
	Min-K%	Token	.702 ± .003	.309 ± .005	.098 ± .004	.025 ± .004
	Min-K%++	Vocab.	.805 ± .002	.531 ± .005	.224 ± .008	.080 ± .012
	RECALL	Seq.	.677 ± .003	.248 ± .005	.038 ± .002	.010 ± .001
	Neighborhood	Seq.	.574 ± .003	.146 ± .004	.015 ± .001	.002 ± .000
	Con-ReCall	Seq.+M	.505 ± .003	.125 ± .003	.013 ± .001	.003 ± .000
	DC-PDD	Token	.653 ± .003	.220 ± .005	.025 ± .002	.001 ± .000
White-box	MoPe	Weights	.802 ± .002	.514 ± .006	.154 ± .006	.023 ± .008
	GradNorm	Grad.	.780 ± .002	.465 ± .006	.141 ± .008	.019 ± .005
Reference-based	DIFF	Seq.	.810 ± .002	.533 ± .005	.235 ± .007	.093 ± .009
	DIFF + DIANA	Vocab.	.918 ± .001	.799 ± .003	.623 ± .005	.470 ± .023
	EZ-MIA	Top-1	.894 ± .002	.755 ± .004	.542 ± .006	.367 ± .013
	EZ-MIA + DIANA	Vocab.	.906 ± .002	.778 ± .003	.602 ± .006	.447 ± .013
	WBC	Token	.894 ± .002	.746 ± .004	.514 ± .007	.341 ± .020
	WBC + DIANA	Vocab.	.919 ± .001	.808 ± .003	.630 ± .007	.488 ± .021
	RATIO	Seq.	.876 ± .002	.690 ± .005	.380 ± .009	.146 ± .018
	RATIO + DIANA	Vocab.	**.925** ± .001	**.824** ± .003	**.678** ± .005	**.542** ± .010

Table 5: **OLMoE on Khan.** Complete results with bootstrap standard deviations.

Type	Method	Access	AUC	TPR at FPR		
				10%	1%	**0.1%**
Text-only	Bag-of-Words	Text	.499 ± .004	.101 ± .004	.008 ± .002	.001 ± .000
Reference-free	Loss	Seq.	.643 ± .004	.227 ± .007	.031 ± .003	.002 ± .001
	ZLIB	Seq.	.622 ± .004	.205 ± .007	.024 ± .003	.004 ± .001
	Lowercase	Seq.	.645 ± .004	.229 ± .006	.029 ± .004	.002 ± .001
	Min-K%	Token	.681 ± .004	.287 ± .007	.055 ± .004	.005 ± .002
	Min-K%++	Vocab.	.767 ± .003	.448 ± .007	.153 ± .007	.054 ± .011
	RECALL	Seq.	.680 ± .004	.279 ± .007	.064 ± .005	.010 ± .002
	Neighborhood	Seq.	.566 ± .004	.139 ± .005	.014 ± .002	.002 ± .001
	Con-ReCall	Seq.+M	.486 ± .004	.127 ± .005	.023 ± .002	.005 ± .001
	DC-PDD	Token	.651 ± .004	.223 ± .007	.002 ± .002	.002 ± .001
White-box	MoPe	Weights	.776 ± .003	.478 ± .007	.174 ± .009	.040 ± .009
	GradNorm	Grad.	.664 ± .004	.275 ± .008	.039 ± .004	.003 ± .001
Reference-based	DIFF	Seq.	.796 ± .003	.502 ± .007	.193 ± .009	.023 ± .026
	DIFF + DIANA	Vocab.	.895 ± .002	.745 ± .006	.518 ± .013	.339 ± .016
	EZ-MIA	Top-1	.883 ± .002	.732 ± .005	.506 ± .009	.338 ± .012
	EZ-MIA + DIANA	Vocab.	.881 ± .002	.727 ± .005	.524 ± .009	.367 ± .020
	WBC	Token	.871 ± .003	.695 ± .006	.467 ± .008	.211 ± .046
	WBC + DIANA	Vocab.	.900 ± .002	.767 ± .005	.561 ± .009	.358 ± .041
	RATIO	Seq.	.849 ± .003	.619 ± .007	.230 ± .015	.033 ± .012
	RATIO + DIANA	Vocab.	**.911** ± .002	**.790** ± .005	**.628** ± .007	**.465** ± .035

Table 6: **OLMoE on Stories.** Complete results with bootstrap standard deviations.

Type	Method	Access	AUC	TPR at FPR		
				10%	1%	**0.1%**
Text-only	Bag-of-Words	Text	.503 ± .003	.099 ± .003	.009 ± .001	.001 ± .000
Reference-free	Loss	Seq.	.663 ± .003	.216 ± .004	.049 ± .002	.014 ± .003
	ZLIB	Seq.	.624 ± .003	.201 ± .004	.029 ± .002	.004 ± .001
	Lowercase	Seq.	.652 ± .003	.257 ± .005	.046 ± .003	.005 ± .001
	Min-K%	Token	.682 ± .003	.249 ± .005	.059 ± .003	.018 ± .003
	Min-K%++	Vocab.	.772 ± .002	.453 ± .005	.155 ± .005	.052 ± .004
	RECALL	Seq.	.636 ± .003	.202 ± .004	.026 ± .002	.005 ± .001
	Neighborhood	Seq.	.587 ± .003	.159 ± .004	.016 ± .002	.002 ± .000
	Con-ReCall	Seq.+M	.499 ± .003	.124 ± .003	.015 ± .001	.003 ± .000
	DC-PDD	Token	.660 ± .003	.242 ± .005	.034 ± .003	.000 ± .001
White-box	MoPe	Weights	.797 ± .002	.493 ± .006	.112 ± .006	.022 ± .006
	GradNorm	Grad.	.686 ± .003	.347 ± .006	.090 ± .006	.010 ± .003
Reference-based	DIFF	Seq.	.793 ± .002	.531 ± .004	.276 ± .008	.136 ± .007
	DIFF + DIANA	Vocab.	.895 ± .002	.756 ± .004	.567 ± .007	.423 ± .016
	EZ-MIA	Top-1	.869 ± .002	.700 ± .004	.472 ± .008	.278 ± .019
	EZ-MIA + DIANA	Vocab.	.886 ± .002	.741 ± .004	.548 ± .006	.391 ± .020
	WBC	Token	.867 ± .002	.693 ± .004	.451 ± .008	.280 ± .013
	WBC + DIANA	Vocab.	.897 ± .002	.760 ± .004	.565 ± .007	.428 ± .013
	RATIO	Seq.	.862 ± .002	.674 ± .005	.408 ± .008	.224 ± .012
	RATIO + DIANA	Vocab.	**.903** ± .002	**.778** ± .004	**.607** ± .007	**.482** ± .018

Table 7: **OLMoE on AutoMath.** Complete results with bootstrap standard deviations.

Type	Method	Access	AUC	TPR at FPR		
				10%	1%	**0.1%**
Text-only	Bag-of-Words	Text	.498 ± .003	.102 ± .003	.010 ± .001	.001 ± .000
Reference-free	Loss	Seq.	.609 ± .003	.174 ± .004	.023 ± .002	.003 ± .001
	ZLIB	Seq.	.589 ± .003	.144 ± .004	.017 ± .001	.002 ± .000
	Lowercase	Seq.	.609 ± .003	.179 ± .004	.028 ± .002	.002 ± .001
	Min-K%	Token	.647 ± .003	.208 ± .005	.031 ± .003	.004 ± .001
	Min-K%++	Vocab.	.747 ± .002	.374 ± .006	.072 ± .004	.013 ± .002
	RECALL	Seq.	.648 ± .003	.214 ± .004	.032 ± .002	.005 ± .001
	Neighborhood	Seq.	.565 ± .003	.146 ± .004	.017 ± .002	.002 ± .001
	Con-ReCall	Seq.+M	.496 ± .003	.116 ± .003	.017 ± .001	.002 ± .001
	DC-PDD	Token	.647 ± .003	.216 ± .004	.030 ± .002	.000 ± .002
White-box	MoPe	Weights	.759 ± .002	.460 ± .005	.158 ± .006	.029 ± .007
	GradNorm	Grad.	.633 ± .003	.234 ± .005	.032 ± .002	.005 ± .001
Reference-based	DIFF	Seq.	.757 ± .002	.452 ± .005	.159 ± .008	.006 ± .004
	DIFF + DIANA	Vocab.	.886 ± .002	.722 ± .004	.462 ± .008	.254 ± .020
	EZ-MIA	Top-1	.873 ± .002	.706 ± .004	.467 ± .007	.285 ± .020
	EZ-MIA + DIANA	Vocab.	.880 ± .002	.722 ± .004	.493 ± .006	.308 ± .018
	WBC	Token	.840 ± .002	.618 ± .004	.343 ± .006	.170 ± .013
	WBC + DIANA	Vocab.	.891 ± .002	.739 ± .004	.483 ± .007	.288 ± .015
	RATIO	Seq.	.842 ± .002	.606 ± .005	.211 ± .008	.025 ± .008
	RATIO + DIANA	Vocab.	**.900** ± .002	**.764** ± .004	**.551** ± .008	**.354** ± .017

Table 8: **OLMoE on Web v2.** Complete results with bootstrap standard deviations.

Type	Method	Access	AUC	TPR at FPR		
				10%	1%	**0.1%**
Text-only	Bag-of-Words	Text	.499 ± .003	.100 ± .003	.009 ± .001	.001 ± .000
Reference-free	Loss	Seq.	.643 ± .003	.233 ± .005	.045 ± .003	.006 ± .002
	ZLIB	Seq.	.594 ± .003	.165 ± .004	.023 ± .002	.003 ± .001
	Lowercase	Seq.	.654 ± .003	.248 ± .004	.050 ± .003	.007 ± .002
	Min-K%	Token	.663 ± .003	.258 ± .005	.054 ± .003	.010 ± .002
	Min-K%++	Vocab.	.776 ± .002	.468 ± .005	.163 ± .007	.047 ± .008
	RECALL	Seq.	.626 ± .003	.206 ± .004	.041 ± .002	.007 ± .001
	Neighborhood	Seq.	.576 ± .003	.156 ± .004	.016 ± .002	.001 ± .000
	Con-ReCall	Seq.+M	.493 ± .003	.122 ± .004	.017 ± .002	.001 ± .000
	DC-PDD	Token	.639 ± .003	.219 ± .005	.033 ± .003	.001 ± .001
White-box	MoPe	Weights	.800 ± .002	.501 ± .006	.166 ± .007	.035 ± .007
	GradNorm	Grad.	.626 ± .003	.267 ± .004	.065 ± .003	.013 ± .002
Reference-based	DIFF	Seq.	.814 ± .002	.570 ± .005	.302 ± .008	.143 ± .014
	DIFF + DIANA	Vocab.	.909 ± .001	.788 ± .003	.620 ± .007	.472 ± .013
	EZ-MIA	Top-1	.891 ± .002	.752 ± .004	.553 ± .007	.385 ± .012
	EZ-MIA + DIANA	Vocab.	.905 ± .002	.781 ± .003	.616 ± .005	.448 ± .013
	WBC	Token	.886 ± .002	.731 ± .004	.512 ± .007	.342 ± .013
	WBC + DIANA	Vocab.	.909 ± .001	.791 ± .003	.622 ± .005	.467 ± .009
	RATIO	Seq.	.867 ± .002	.673 ± .004	.380 ± .009	.193 ± .015
	RATIO + DIANA	Vocab.	**.916** ± .001	**.808** ± .003	**.661** ± .006	**.528** ± .013

Table 9: **OLMoE on WikiHow.** Complete results with bootstrap standard deviations.

Type	Method	Access	AUC	TPR at FPR		
				10%	1%	**0.1%**
Text-only	Bag-of-Words	Text	.503 ± .003	.102 ± .003	.010 ± .001	.001 ± .000
Reference-free	Loss	Seq.	.689 ± .003	.279 ± .006	.058 ± .003	.012 ± .003
	ZLIB	Seq.	.628 ± .003	.201 ± .005	.026 ± .002	.002 ± .000
	Lowercase	Seq.	.698 ± .003	.299 ± .005	.052 ± .004	.001 ± .001
	Min-K%	Token	.723 ± .003	.339 ± .006	.077 ± .004	.021 ± .002
	Min-K%++	Vocab.	.798 ± .002	.501 ± .006	.185 ± .007	.053 ± .006
	RECALL	Seq.	.669 ± .003	.249 ± .005	.035 ± .003	.006 ± .001
	Neighborhood	Seq.	.559 ± .003	.129 ± .003	.014 ± .001	.001 ± .000
	Con-ReCall	Seq.+M	.474 ± .003	.105 ± .003	.015 ± .001	.002 ± .001
	DC-PDD	Token	.660 ± .003	.220 ± .005	.001 ± .000	.001 ± .000
White-box	MoPe	Weights	.804 ± .002	.518 ± .006	.186 ± .007	.036 ± .009
	GradNorm	Grad.	.736 ± .003	.414 ± .006	.106 ± .005	.021 ± .004
Reference-based	DIFF	Seq.	.808 ± .002	.519 ± .007	.175 ± .007	.055 ± .006
	DIFF + DIANA	Vocab.	.904 ± .001	.775 ± .003	.576 ± .007	.406 ± .017
	EZ-MIA	Top-1	.891 ± .002	.744 ± .004	.494 ± .008	.284 ± .026
	EZ-MIA + DIANA	Vocab.	.897 ± .002	.761 ± .003	.563 ± .007	.395 ± .013
	WBC	Token	.876 ± .002	.696 ± .005	.406 ± .007	.204 ± .013
	WBC + DIANA	Vocab.	.900 ± .002	.772 ± .004	.555 ± .008	.354 ± .022
	RATIO	Seq.	.830 ± .002	.559 ± .005	.171 ± .008	.049 ± .008
	RATIO + DIANA	Vocab.	**.911** ± .001	**.793** ± .003	**.618** ± .006	**.473** ± .013

F Additional Targets, Scale, References, and Defenses

Cross-target operating points. The following table expands Table 2 with WBC, all four metrics, and sequence-bootstrap standard deviations. Each target uses its own pre-trained base reference; the Stanford model comparisons and three additional OLMoE corpora use the training protocols in Appendix D.

With 10k–20k non-members, 1% FPR permits 100–200 false positives, versus 10–20 at 0.1%. At 0.1% FPR, RATIO+DIANA has lower TPR than WBC on Qwen2.5-1.5B, CC-News and Wikipedia and RATIO on CC-News.

Table 10: **Cross-target performance at all operating points.** Estimate ± sequence-bootstrap standard deviation. +DIANA uses RATIO aggregation.

	AUC			**TPR at 10% FPR**		
Target / corpus	RATIO	WBC	+DIANA	RATIO	WBC	+DIANA
Stanford: target model						
OLMoE-1B-7B	.876 ± .002	.894 ± .002	**.925** ± .001	.690 ± .005	.746 ± .004	**.824** ± .003
Llama-3.1-8B	.939 ± .001	.947 ± .001	**.960** ± .001	.854 ± .003	.877 ± .003	**.912** ± .002
PYTHIA-2.8B	.765 ± .002	.825 ± .002	**.864** ± .002	.409 ± .006	.576 ± .005	**.679** ± .004
PYTHIA-1.4B	.740 ± .002	.811 ± .002	**.850** ± .002	.353 ± .005	.535 ± .005	**.641** ± .005
OPT-1.3B	.708 ± .003	.800 ± .002	**.837** ± .002	.296 ± .006	.506 ± .005	**.611** ± .005
Qwen2.5-1.5B	.709 ± .003	**.788** ± .002	.780 ± .002	.295 ± .006	**.441** ± .006	.429 ± .006
Mamba-1.4B	.735 ± .002	.821 ± .002	**.859** ± .002	.313 ± .005	.514 ± .006	**.637** ± .006
OLMoE*: non-generated corpus*						
XSum	.917 ± .002	.966 ± .001	**.967** ± .001	.832 ± .005	.927 ± .003	**.927** ± .003
CC-News	.875 ± .003	**.934** ± .002	.928 ± .002	.616 ± .016	**.899** ± .006	.890 ± .008
Wikipedia	.926 ± .002	**.974** ± .001	.972 ± .001	.824 ± .006	**.943** ± .002	.938 ± .002

	TPR at 1% FPR			**TPR at 0.1% FPR**		
Target / corpus	RATIO	WBC	+DIANA	RATIO	WBC	+DIANA
Stanford: target model						
OLMoE-1B-7B	.380 ± .009	.514 ± .007	**.678** ± .005	.146 ± .018	.341 ± .020	**.542** ± .010
Llama-3.1-8B	.696 ± .007	.765 ± .005	**.851** ± .003	.512 ± .013	.661 ± .016	**.799** ± .006
PYTHIA-2.8B	.099 ± .006	.243 ± .009	**.406** ± .008	.018 ± .004	.097 ± .013	**.233** ± .019
PYTHIA-1.4B	.074 ± .004	.205 ± .006	**.357** ± .007	.010 ± .002	.059 ± .011	**.187** ± .012
OPT-1.3B	.058 ± .004	.184 ± .007	**.302** ± .008	.007 ± .002	.053 ± .007	**.136** ± .012
Qwen2.5-1.5B	.045 ± .003	**.120** ± .005	.096 ± .005	.007 ± .001	**.034** ± .005	.014 ± .003
Mamba-1.4B	.055 ± .004	.161 ± .007	**.276** ± .008	.007 ± .002	.027 ± .007	**.089** ± .015
OLMoE*: non-generated corpus*						
XSum	.021 ± .002	.773 ± .012	**.835** ± .006	.008 ± .002	.597 ± .016	**.691** ± .031
CC-News	.017 ± .002	.029 ± .008	**.039** ± .008	**.004** ± .001	.002 ± .001	.001 ± .001
Wikipedia	.251 ± .025	**.893** ± .004	.879 ± .004	.019 ± .008	**.836** ± .007	.781 ± .033

Weaker fine-tuning. In an earlier training configuration, smaller dense models were evaluated after one epoch at 5×10^{-6}. PYTHIA-1.4B gives AUC 0.516, 0.519, and 0.522 for RATIO, WBC, and RATIO+DIANA. These attacks have little discriminatory power in this setting. Chance-level attack performance alone does not establish the absence of memorization.

Training duration. Table 11 reports the first five epochs of the OLMoE–Stanford checkpoint ladder at learning rate 2×10^{-6}, with three training seeds and 1,500 members plus 1,500 non-members per seed. At epoch one, RATIO+DIANA catches 30.6% of members at 0.1% FPR, compared with 10.8% for WBC. By epoch four, all three attacks exceed 99.4% TPR. This ladder uses the same fixed $\tau = 1$ as the primary comparisons. Appendix N examines the centered gains in a twenty-epoch extension.

Table 11: **Training-duration sweep, epochs 1–5.** OLMoE on Stanford; mean ± sample standard deviation across three training seeds. +DIANA uses RATIO aggregation and $\tau = 1$.

	AUC			TPR at 0.1% FPR		
Epoch	RATIO	WBC	+DIANA	RATIO	WBC	+DIANA
1	.862 ± .001	.895 ± .001	**.944** ± .001	.027 ± .001	.108 ± .021	**.306** ± .011
2	.985 ± .000	.993 ± .000	**.998** ± .000	.327 ± .005	.583 ± .031	**.901** ± .003
3	.999 ± .000	1.000 ± .000	**1.000** ± .000	.898 ± .003	.875 ± .021	**.996** ± .000
4	1.000 ± .000	1.000 ± .000	**1.000** ± .000	.995 ± .000	.997 ± .001	**1.000** ± .000
5	1.000 ± .000	1.000 ± .000	**1.000** ± .000	1.000 ± .000	1.000 ± .000	**1.000** ± .000

Target and reference sizes. Table 12 compares PYTHIA targets from 160M to 6.9B on Stanford, each using its own pre-trained base and its training recipe in Appendix D. Table 13 fixes the 2.8B target model and target sequences and varies only the reference. All references share the PYTHIA tokenizer; the moments are recomputed for each pair, with $\tau = 1$.

Across target sizes, DIANA improves AUC over RATIO for every nonsaturated target; all three attacks round to 1.000 at 6.9B. At 1% FPR, calibration raises TPR from 0.092 to 0.411 on PYTHIA-410M and from 0.074 to 0.357 on PYTHIA-1.4B. With the 2.8B target fixed, DIANA has the highest AUC at every tested reference size. Its AUC is 0.715 with a 160M reference and 0.864 with the target's true base.

Table 12: **Target-size sweep.** PYTHIA on Stanford; +DIANA denotes RATIO+DIANA. Target size varies, and every target uses its own pre-trained base.

	AUC			TPR at 1% FPR		
Target size	RATIO	WBC	+DIANA	RATIO	WBC	+DIANA
160M	.504	**.506**	.504	.009	.009	**.011**
410M	.775	.844	**.877**	.092	.260	**.411**
1B	.577	.613	**.629**	.017	.025	**.030**
1.4B	.740	.811	**.850**	.074	.205	**.357**
2.8B	.765	.825	**.864**	.099	.243	**.406**
6.9B	1.000	1.000	**1.000**	1.000	1.000	**1.000**

Table 13: **Reference-size sweep.** Fixed PYTHIA-2.8B Stanford target; the row label is reference size. +DIANA denotes RATIO+DIANA.

	AUC			TPR at 1% FPR		
Reference size	RATIO	WBC	+DIANA	RATIO	WBC	+DIANA
160M	.695	.687	**.715**	.060	.058	**.067**
410M	.749	.761	**.795**	.086	.127	**.173**
1B	.761	.781	**.816**	.098	.161	**.241**
1.4B	.762	.791	**.822**	.098	.167	**.254**
2.8B *(true base)*	.765	.825	**.864**	.099	.243	**.406**
6.9B	.747	.770	**.796**	.083	.132	**.193**

LoRA. The LoRA comparison uses PYTHIA-2.8B on Stanford for two epochs, with ranks 16 and 64 at learning rate 10^{-4}, evaluated on 20,000 target sequences per class. The full fine-tuning comparator reuses the one-epoch target from the transfer and size comparisons, trained at 5×10^{-6}. The LoRA runs use batch size 16, weight decay 0.1, and 500 warmup steps; LoRA applies to the query–key–value projections with scale parameter $2r$ and dropout 0.05. The comparison evaluates these training recipes jointly, with different learning rates, durations, and update parameterizations. DIANA leads

AUC and TPR at 1% FPR for all three recipes: its AUC is 0.864 for full fine-tuning, 0.732 for rank 64, and 0.607 for rank 16 (Table 14).

DP-SGD. The DP-SGD comparison uses PYTHIA-410M on Khan Academy for three epochs at learning rate 5×10^{-5}, with expected batch size 16, clipping norm 1, and $\delta = 10^{-5}$, evaluated on 10,000 target sequences per class. Opacus uses Poisson sampling and an RDP accountant. The stored final privacy budgets are 99,999.996, 7.9959, and 3.9955, with Gaussian noise multipliers 0.0666, 0.4553, and 0.5603, respectively. The high-budget row therefore includes nonzero noise. The no-DP row uses ordinary AdamW with the same learning rate, batch size, model, corpus, and epoch count. Without DP, DIANA reaches 1.000 AUC. At $\varepsilon \approx 8$ and 4, all three attacks have AUC within 0.005 of 0.5; even the high-budget, low-noise run reduces their AUC to 0.527–0.536. Table 14 supplies all four metrics and their recorded uncertainty.

Table 14: **Complete defense results.** Estimate ± recorded sequence-bootstrap standard deviation. +DIANA denotes RATIO+DIANA, with $\tau = 1$. LoRA rows use PYTHIA-2.8B on Stanford; DP rows use PYTHIA-410M on Khan.

	AUC			TPR at 10% FPR		
Training recipe	RATIO	WBC	+DIANA	RATIO	WBC	+DIANA
Full fine-tuning	.765 ± .002	.825 ± .002	**.864** ± .002	.409 ± .006	.576 ± .005	**.679** ± .004
LoRA, $r = 64$	.644 ± .003	.685 ± .003	**.732** ± .003	.211 ± .005	.282 ± .005	**.349** ± .005
LoRA, $r = 16$	.569 ± .003	.581 ± .003	**.607** ± .003	.145 ± .004	.163 ± .004	**.186** ± .005
No DP	.999 ± .000	.999 ± .000	**1.000** ± .000	1.000 ± .000	.997 ± .001	**1.000** ± .000
$\varepsilon \approx 10^5$	.527 ± .004	**.536** ± .004	.534 ± .004	.116 ± .005	**.128** ± .005	.127 ± .004
$\varepsilon \approx 8$	.505 ± .004	**.505** ± .004	.503 ± .004	.100 ± .004	**.106** ± .005	.104 ± .005
$\varepsilon \approx 4$	.503 ± .004	**.504** ± .004	.502 ± .004	.102 ± .004	**.105** ± .004	.104 ± .004
	TPR at 1% FPR			**TPR at 0.1% FPR**		
Training recipe	RATIO	WBC	+DIANA	RATIO	WBC	+DIANA
Full fine-tuning	.099 ± .006	.243 ± .009	**.406** ± .008	.018 ± .004	.097 ± .013	**.233** ± .019
LoRA, $r = 64$	.034 ± .003	.057 ± .003	**.077** ± .004	.005 ± .001	.010 ± .002	**.018** ± .002
LoRA, $r = 16$	.018 ± .002	.022 ± .002	**.026** ± .002	.002 ± .001	.004 ± .001	**.004** ± .001
No DP	.996 ± .001	.990 ± .001	**.999** ± .000	.908 ± .060	.974 ± .004	**.996** ± .001
$\varepsilon \approx 10^5$	.011 ± .001	.015 ± .002	**.015** ± .002	.002 ± .001	.002 ± .001	**.002** ± .001
$\varepsilon \approx 8$	.011 ± .002	.010 ± .001	**.011** ± .001	.001 ± .001	.001 ± .000	**.001** ± .000
$\varepsilon \approx 4$	**.010** ± .002	.010 ± .001	.009 ± .001	.001 ± .000	.001 ± .001	**.001** ± .000

G PER-CORPUS COMPONENT AND GAIN ABLATIONS

Table 15 expands the first group of Table 3. Centering improves AUC and TPR on all six corpora, and adding the regularized variance weight improves both metrics further on every corpus. Weighting the observed gain increases AUC everywhere but increases TPR at 0.1% FPR on five of six corpora. Table 16 expands the denominator comparison in the second group of Table 3.

Using more of the null distribution. Define the upper-tail probability $r_j = \sum_{v \in \mathcal{V}} p_j^{\mathrm{T}}(v) \mathbf{1}\{G_j(v) \geq G_j^{\mathrm{obs}}\}$. We compare aggregating $-\log r_j$ and $1/2 - r_j$ with centered gains and a Cornish–Fisher skewness correction (Table 17). The full-null transformations attain lower AUC than the calibrated gain with $\tau = 1$. This comparison concerns those transformations and their sequence aggregation; access to the full null distribution does not itself impose an information loss.

Table 15: **Centering and weighting by corpus.** RATIO aggregation; weight $1/(1+\sigma_j^2)$ when present.

Component	Stanford	Khan	Stories	AutoMath	Web v2	WikiHow
AUC						
Observed gain	.876	.849	.862	.842	.867	.830
Weight only	.881	.867	.883	.866	.891	.851
Center only	.915	.902	.896	.887	.906	.897
DIANA	**.925**	**.911**	**.903**	**.900**	**.916**	**.911**
TPR at 0.1% *FPR*						
Observed gain	.146	.033	.224	.025	.193	.049
Weight only	.057	.036	.253	.044	.219	.049
Center only	.489	.391	.443	.301	.509	.374
DIANA	**.542**	**.465**	**.482**	**.354**	**.528**	**.473**

Table 16: **Denominator shapes by corpus.** RATIO aggregation on OLMOE; μ_j and σ_j^2 are the target-weighted gain mean and variance. All numerators are centered gains.

Weight on s_j	Stanford	Khan	Stories	AutoMath	Web v2	WikiHow
AUC						
1 (center only)	.915	.902	.896	.887	.906	.897
$1/(1+\sigma_j)$	.924	.910	.902	.897	.914	.908
$1/\sqrt{1+\sigma_j^2}$	.924	.910	.901	.896	.913	.908
$1/(1+\sigma_j^2)$	**.925**	**.911**	**.903**	**.900**	**.916**	**.911**
$1/\sigma_j^2$	.847	.810	.827	.798	.850	.832
TPR at 1% *FPR*						
1 (center only)	.641	.597	.581	.506	.632	.556
$1/(1+\sigma_j)$	.675	.622	.606	.546	.659	.610
$1/\sqrt{1+\sigma_j^2}$	.675	.628	.605	.546	.658	.611
$1/(1+\sigma_j^2)$	**.678**	**.628**	**.607**	**.551**	**.661**	**.618**
$1/\sigma_j^2$	.037	.018	.032	.016	.036	.028
TPR at 0.1% *FPR*						
1 (center only)	.489	.391	.443	.301	.509	.374
$1/(1+\sigma_j)$	.534	.438	.475	.353	.529	.450
$1/\sqrt{1+\sigma_j^2}$	.538	.452	.481	.353	**.532**	.447
$1/(1+\sigma_j^2)$	**.542**	**.465**	**.482**	**.354**	.528	**.473**
$1/\sigma_j^2$	.002	.002	.001	.001	.002	.002

Table 17: **Alternative token transformations.** AUC with RATIO aggregation on OLMOE. The center-plus-variance row uses $\tau = 1$.

Token transformation	Stanford	Khan	Stories	AutoMath	Web v2	WikiHow
Center only	.915	.902	.896	.887	.906	.897
Center + variance	**.925**	**.911**	**.903**	**.900**	**.916**	**.911**
Cornish–Fisher	.522	.535	.518	.534	.518	.530
Full null: $-\log r_j$	.834	.800	.834	.822	.861	.798
Full null: $1/2 - r_j$	.589	.545	.609	.558	.613	.587

Table 18: **Calibration components on PYTHIA-2.8B.** AUC with RATIO aggregation at epoch two. The variance weight uses $\tau = 1$.

Calibration	Stanford	Khan	Stories	AutoMath	Web v2	WikiHow
Center only	.733	.834	.693	.694	.729	.687
Variance, $\tau = 1$	**.757**	**.862**	**.710**	**.718**	**.750**	**.714**

H ALTERNATIVE CENTERING TERMS

The centering term as a KL divergence. The log-gain function in Equation (1) is $G_j(v) = \log[p_j^{\mathrm{T}}(v)/p_j^{\mathrm{R}}(v)]$. Substituting this log-ratio into the target-weighted mean gives $\mu_j = \sum_{v\in\mathcal{V}} p_j^{\mathrm{T}}(v) \log[p_j^{\mathrm{T}}(v)/p_j^{\mathrm{R}}(v)] = D_{\mathrm{KL}}(p_j^{\mathrm{T}} \| p_j^{\mathrm{R}})$, the KL divergence between the target and reference predictions, so the calibrated gain in Equation (5) can be written as

$$\tilde{s}_j = \frac{G_j^{\mathrm{obs}} - D_{\mathrm{KL}}(p_j^{\mathrm{T}} \| p_j^{\mathrm{R}})}{\tau + \sigma_j^2}.$$

The centering term is therefore the expected gain that fine-tuning adds to a next token drawn from the target's own prediction.

Comparison with other centering terms. This forward KL is the centering term used by DIANA. Table 19 compares subtracting this mean with subtracting reverse KL, the average of the two KLs, and changes in model confidence. Each term is evaluated with RATIO, DIFF, and WBC on the same $5{,}000$ members and $5{,}000$ non-members per corpus, without variance weighting. Higher scores indicate membership for every variant.

The confidence changes are target-minus-reference differences in entropy, log top-1 probability, and the observed token's logit margin over its strongest competitor. Both signs of the entropy change are included. Forward-KL centering has the highest AUC on all six corpora with RATIO and WBC, and on four with DIFF. Averaging the two KLs gives slightly higher DIFF AUC on Stories and Web Samples v2. These comparisons distinguish the exact centering identity from the empirical ranking of alternatives under a particular aggregation.

Table 19: **Centering-term alternatives.** AUC on six OLMoE corpora, with 5,000 target sequences per class. Every aggregation uses the same alternatives and target sequences; all rows omit variance weighting.

Centering term	Stanford	Khan	Stories	AutoMath	Web v2	WikiHow
Ratio aggregation						
Observed gain	.875	.844	.860	.840	.861	.826
Forward KL	**.911**	**.904**	**.894**	**.888**	**.899**	**.896**
Reverse KL	.843	.805	.878	.840	.881	.836
Mean of KLs	.887	.867	.890	.871	.894	.882
Subtract entropy change	.751	.696	.755	.724	.758	.713
Add entropy change	.795	.750	.802	.785	.796	.811
Subtract top-1 change	.883	.866	.865	.857	.865	.859
Subtract margin change	.395	.415	.370	.392	.390	.424
Diff aggregation						
Observed gain	.808	.789	.785	.757	.807	.807
Forward KL	**.901**	**.887**	.884	**.872**	.890	**.889**
Reverse KL	.863	.821	.882	.843	.884	.849
Mean of KLs	.892	.866	**.887**	.865	**.892**	.886
Subtract entropy change	.699	.703	.702	.697	.707	.699
Add entropy change	.798	.751	.801	.753	.799	.813
Subtract top-1 change	.838	.796	.809	.763	.835	.848
Subtract margin change	.343	.376	.324	.349	.330	.398
WBC aggregation						
Observed gain	.889	.870	.864	.839	.881	.874
Forward KL	**.908**	**.898**	**.892**	**.886**	**.898**	**.892**
Reverse KL	.891	.851	.882	.866	.893	.882
Mean of KLs	.901	.875	.890	.878	.897	.892
Subtract entropy change	.777	.772	.775	.770	.778	.767
Add entropy change	.780	.716	.782	.731	.776	.803
Subtract top-1 change	.862	.789	.838	.756	.862	.871
Subtract margin change	.259	.295	.269	.289	.259	.264

I ROBUSTNESS TO τ

The hyperparameter τ in $s_j/(\tau + \sigma_j^2)$ can be fixed directly, without an unlabeled pool. Every tested $\tau \in \{0.25, 0.5, 1, 2, 4\}$ lies within 0.005 AUC of the best tested value on each corpus (Figure 5). Across the full positive sweep, $\tau \in \{0.1, 0.25, 0.5, 1, 2, 4, 8, 16\}$, all six corpora remain above unadjusted RATIO. With $\tau = 0$, AUC falls to 0.798–0.850: a positive variance floor prevents nearly zero variances from receiving excessive weight, while a broad range of fixed values performs similarly.

We therefore use the same fixed $\tau = 1$ in the primary, transfer, and supporting evaluations (Appendix D). No estimate from the target-sequence pool or model-specific adjustment is needed.

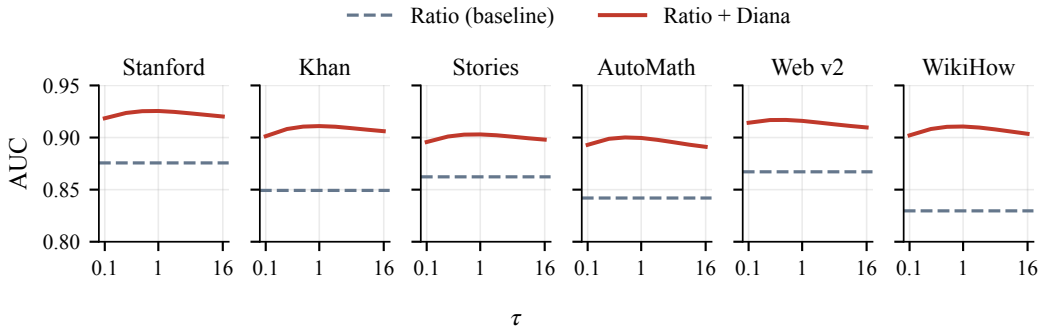


Figure 5: **Sensitivity to τ by corpus.** The curves from Figure 4b with uncalibrated RATIO baselines, on shared axes. A broad range of fixed τ values preserves the calibration gain; no pool estimate is used in this sweep.

J COMPARISON WITH LEARNED CALIBRATIONS

DIANA computes its calibration in closed form from the two models' predictions, without membership labels. A natural alternative is to learn a per-token calibration from labeled members and non-members. This appendix tests whether such supervised calibrations improve on the closed form when they are given membership labels and additional per-token features. None does: the best learned calibration reaches a mean TPR of 0.457 at 0.1% FPR, compared with 0.474 for RATIO+DIANA on the same target sequences (Table 20).

Density-ratio classifier. This baseline learns a flexible per-token score. For each token, it uses twelve features: the centered gain, the target-weighted mean gain, the log gain variance, the target and reference log-probabilities, the reverse KL divergence, the changes in entropy, top-1 probability, margin, and rank between the two models, the log χ^2 divergence, and the relative position. A gradient-boosted classifier (300 iterations, learning rate 0.1, 63 leaves) predicts membership from these features, and the attack score sums its per-token logits over the sequence.

Gaussian discriminant fits. These baselines keep DIANA's centered gain as the only token-level input but learn how to weight it. Tokens are grouped into bins by their log gain variance and target-weighted mean gain. Within each bin, the member and non-member centered gains are modeled as Gaussians with fitted means. Linear discriminant analysis (LDA) assumes one shared variance per bin, so the per-token log-likelihood ratio is linear in the centered gain, with a bin-dependent slope and intercept. The fitted slope plays the role of DIANA's weight $1/(\tau + \sigma_j^2)$. Quadratic discriminant analysis (QDA) fits separate member and non-member variances.

Rows of Table 20.

- *Density ratio, in-domain / LODO*: the classifier above, trained either on half of the target sequences from the same corpus or on the other five corpora.
- *LDA slope, sum*: sums only the slope term of the LDA log-likelihood ratio over tokens.

- *LDA + offset, sum*: also adds the bin-dependent intercept.
- *QDA, sum*: sums the QDA log-likelihood ratio.
- *LDA + offset,* RATIO: replaces the sum with RATIO aggregation of the same per-token LDA outputs, matching the aggregation used by the comparator.

Training and comparison protocol. All learned calibrations use membership labels, which DIANA does not require. The in-domain density-ratio classifier uses a sequence-level, membership-stratified $50/50$ train/test split within each corpus. All other rows use leave-one-domain-out (LODO) evaluation: they are fit on five corpora and tested on the sixth. Each learned calibration is compared with RATIO+DIANA ($\tau = 1$) on exactly the same test sequences. Because the in-domain split uses a different test subset, its comparator differs slightly (0.477 instead of 0.474).

Results. Every learned calibration has lower mean AUC and lower mean TPR at 0.1% FPR than its matched DIANA comparator. The density-ratio classifier, despite its twelve features, reaches only 0.331 in-domain and 0.349 under LODO. Among the Gaussian fits, adding the intercept improves LDA from 0.392 to 0.425, and RATIO aggregation raises it to 0.457, the closest result to DIANA. QDA performs worst, at 0.115. In these tests, labeled fitting does not recover more membership evidence from the per-token features than the closed-form calibration does without labels.

Table 20: **Learned calibrations versus DIANA.** Means over six corpora. Each learned calibration uses membership labels; its matched comparator, RATIO+DIANA with $\tau = 1$, uses none and is evaluated on the same test sequences. LODO: fit on five corpora, tested on the sixth.

	Learned calibration		**Matched DIANA variant**	
Method / aggregation	AUC	TPR at 0.1% FPR	AUC	TPR at 0.1% FPR
Density ratio, in-domain	.885	.331	**.911**	**.477**
Density ratio, LODO	.889	.349	**.911**	**.474**
LDA slope, sum	.900	.392	**.911**	**.474**
LDA + offset, sum	.906	.425	**.911**	**.474**
QDA, sum	.792	.115	**.911**	**.474**
LDA + offset, RATIO	.903	.457	**.911**	**.474**

K TOP-k PROBABILITY EVALUATION

For each position, we take the union of the two models' top-k token sets. We retain *both* models' probabilities for every token in this union and renormalize each distribution on that support. Their renormalized gain mean and variance, denoted $\mu_{j,k}$ and $\sigma^2_{j,k}$, replace the full-vocabulary moments, while the observed-token gain G_j^{obs} is kept exact. The DIFF+DIANA attack score with top-k moments is the sequence mean of $(G_j^{\text{obs}} - \mu_{j,k})/(1 + \sigma^2_{j,k})$: DIFF aggregation with $\tau = 1$. Its full-vocabulary counterpart uses the same aggregation and τ. Unadjusted DIFF provides the matched baseline on the same target sequences; Table 21 also includes RATIO.

This experiment requires cross-model probabilities on the union, as well as the observed token's probability even when it is outside that union. Two independent top-k lists from a conventional API need not expose these quantities. It therefore evaluates how well top-k probabilities approximate the full-vocabulary moments under that access assumption.

Even at $k = 5$, DIFF+DIANA exceeds both unadjusted attacks in TPR at 0.1% FPR on every corpus, reaching 30.3% on average against unadjusted DIFF's 7.6%. At $k = 50$, it retains 90%–96% of its full-vocabulary TPR, with mean 37.1% versus 39.4% (Figure 6 and table 21). Restricting the moments to the top-k union changes the centering term as well as the weight, so these outcomes reflect both approximations.

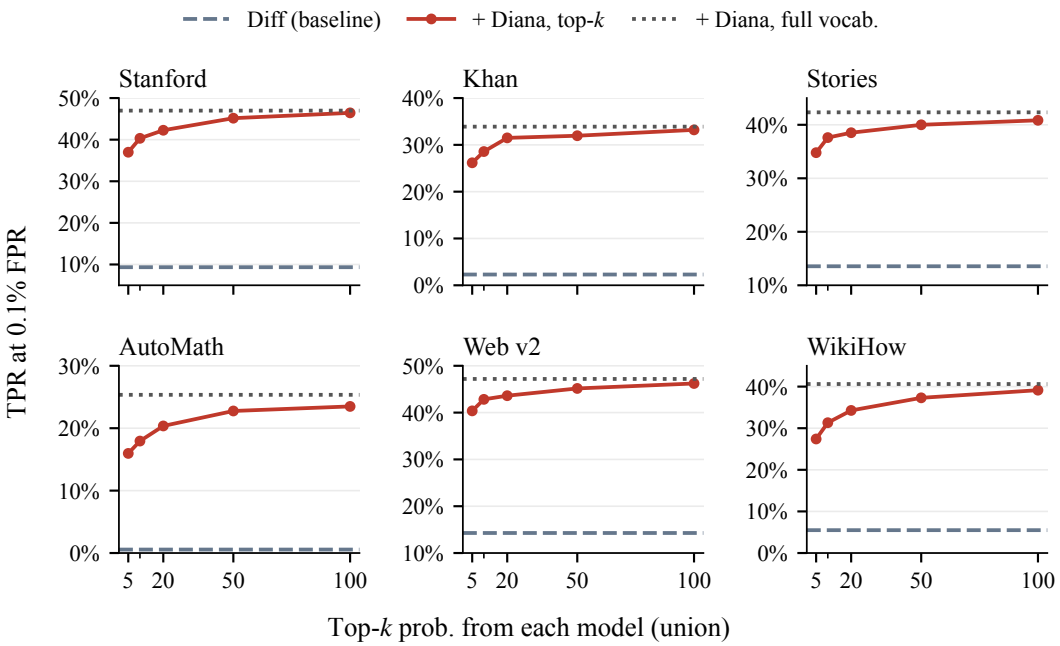


Figure 6: **Top-k moments by corpus.** DIFF with and without DIANA ($\tau = 1$); the dotted line uses full-vocabulary moments. Budgets $k \in \{5, 10, 20, 50, 100\}$ use a linear axis; TPR scales vary by corpus.

Table 21: **Top-k calibration and unadjusted baselines.** DIANA uses exact observed gain, moments on the renormalized support union, $\tau = 1$, and DIFF aggregation. Budgets are per model.

Attack / budget	Stanford	Khan	Stories	AutoMath	Web v2	WikiHow
AUC						
DIFF (raw)	.810	.796	.793	.757	.814	.808
RATIO (raw)	.876	.849	.862	.842	.867	.830
Top-5	.902	.882	.881	.863	.894	.885
Top-10	.908	.889	.886	.874	.900	.892
Top-20	.912	.892	.889	.880	.904	.896
Top-50	.915	.893	.892	.883	.906	.900
Top-100	.916	.894	.893	.884	.907	.901
DIANA, full	**.918**	**.895**	**.895**	**.886**	**.909**	**.903**
TPR at 0.1% *FPR*						
DIFF (raw)	.093	.023	.136	.006	.143	.055
RATIO (raw)	.146	.033	.224	.025	.193	.049
Top-5	.370	.262	.348	.160	.404	.274
Top-10	.403	.286	.376	.179	.428	.313
Top-20	.423	.315	.385	.204	.436	.343
Top-50	.452	.320	.400	.228	.452	.373
Top-100	.464	.332	.408	.235	.462	.391
DIANA, full	**.470**	**.339**	**.423**	**.254**	**.472**	**.406**

L UTILITY AND GENERALIZATION OF THE EVALUATED TARGETS

This appendix characterizes every evaluated target checkpoint by its perplexity on member and held-out text, before and after fine-tuning (Table 22). Two summaries are used throughout. The *gain* is the reduction in held-out perplexity from the pre-trained reference to the target; it measures how much the target has learned about the domain. The *loss gap* is the target's held-out minus member negative log-likelihood per token; it measures how much better the target fits its training sequences than new text from the same domain. A realistic fine-tuning target should show a clear gain, and the loss gap indicates how much membership signal is available. In every row, the reference model's member and held-out perplexities are nearly identical, so differences after fine-tuning come from training.

Primary OLMoE targets. On the six Cosmopedia corpora, fine-tuning reduces held-out perplexity by 27–37% and member perplexity by 34–45%, with loss gaps of 0.108–0.130 nats per token. The targets therefore adapt substantially to the domain while fitting their training sequences more closely than held-out text. On the three non-generated corpora, the gain is only 6.9–7.2%, but the loss gaps are similar or slightly larger (0.132–0.141).

Other model families. The one-epoch Llama-3.1-8B, PYTHIA-2.8B, OPT-1.3B, and Mamba-1.4B targets reduce held-out perplexity by 28.5–37.0%, with loss gaps of 0.064–0.195. Qwen2.5-1.5B is the exception: its gain is 4.9% and its loss gap is 0.013. This small change is consistent with its having the smallest TPR improvement among the target models in Table 2.

Target size. The PYTHIA size comparison (Appendix F) includes checkpoints with very different loss gaps. The 160M and 1B targets have gaps of 0.001 and 0.017; attacks are at chance on the 160M target and weak on the 1B target (AUC at most 0.629). The 410M, 1.4B, and 2.8B targets have gains of 36.3–39.4% and gaps of 0.074–0.106. The 6.9B target has a gap of 0.750, and all attacks on it are saturated.

LoRA and DP-SGD. LoRA with ranks 64 and 16 keeps most of the gain of full fine-tuning (34.7% and 32.7%) but lowers the loss gap to 0.039 and 0.017, matching the weaker attacks on these targets. The DP-SGD rows use PYTHIA-410M on Khan Academy. Without DP, the loss gap is 0.844. With DP-SGD, it falls to 0.013 at $\varepsilon \approx 10^5$ and to 0.007 at $\varepsilon \approx 8$ and 4, where all attacks are close to chance. At these two budgets, the gain also falls to 4.8% and 3.4%, so the privacy protection comes with a large loss of utility.

Table 22: **Utility and generalization of every evaluated target.** Perplexity (PPL) on member and held-out text for the pre-trained reference $\mathcal{M}^{\mathrm{R}}$ and the fine-tuned target $\mathcal{M}^{\mathrm{T}}$. Gain is the held-out perplexity reduction; loss gap is the target's held-out minus member NLL (nats/token).

Corpus / target	Member PPL		Held-out PPL		Generalization	
	$\mathcal{M}^{\mathrm{R}}$	$\mathcal{M}^{\mathrm{T}}$	$\mathcal{M}^{\mathrm{R}}$	$\mathcal{M}^{\mathrm{T}}$	Gain (%)	Loss gap
OLMoE-1B-7B: corpus						
Stanford	4.92	2.82	4.91	3.20	34.9	0.126
Khan	3.66	2.40	3.69	2.70	26.9	0.117
Stories	6.97	3.83	6.97	4.37	37.3	0.130
AutoMath	5.21	3.27	5.22	3.64	30.2	0.108
Web v2	6.14	3.71	6.10	4.22	30.9	0.127
WikiHow	4.58	2.55	4.58	2.87	37.4	0.116
XSum	8.67	7.07	8.67	8.07	6.9	0.132
CC-News	8.69	6.99	8.67	8.04	7.2	0.140
Wikipedia	8.22	6.62	8.20	7.63	7.0	0.141
Stanford: target model						
Llama-3.1-8B	3.83	2.25	3.82	2.74	28.5	0.195
Pythia-160M	25.72	20.04	25.71	20.05	22.0	0.001
Pythia-410M	10.33	5.63	10.33	6.26	39.4	0.106
Pythia-1B	8.47	5.86	8.47	5.96	29.6	0.017
Pythia-1.4B	7.74	4.58	7.74	4.93	36.3	0.074
Pythia-2.8B	6.87	3.99	6.87	4.33	37.0	0.080
Pythia-6.9B	6.50	1.97	6.49	4.16	35.9	0.750
OPT-1.3B	8.65	5.16	8.64	5.51	36.2	0.066
Qwen2.5-1.5B	3.87	3.63	3.87	3.68	4.9	0.013
Mamba-1.4B	7.03	4.27	7.02	4.56	35.1	0.064
Training variants: Stanford (LoRA), Khan (DP)						
Pythia-2.8B, LoRA 64	6.87	4.32	6.87	4.49	34.7	0.039
Pythia-2.8B, LoRA 16	6.87	4.55	6.87	4.63	32.7	0.017
Pythia-410M, no DP	7.30	2.04	7.35	4.74	35.5	0.844
Pythia-410M, high budget	7.30	6.01	7.35	6.09	17.1	0.013
Pythia-410M, DP 8	7.30	6.95	7.35	7.00	4.8	0.007
Pythia-410M, DP 4	7.30	7.05	7.35	7.10	3.4	0.007

M EMPIRICAL CHECKS OF THE CALIBRATION

The calibrated gain $\tilde{s}_j = s_j/(\tau + \sigma_j^2)$ relies on two quantities that DIANA computes from the trained target: the target-weighted mean gain μ_j, which should match the average non-member gain (Proposition 1), and the variance $\hat{\sigma}_j^2$, which should describe how widely non-member centered gains fluctuate at each prefix. This appendix checks both on held-out text from all six corpora. Appendix M.1 extends the non-member check of Section 3.3 and confirms that members retain a higher mean. Appendix M.2 checks the variance and uses the results to motivate the form of the denominator $\tau + \hat{\sigma}_j^2$.

Protocol. For each corpus, we sample $5{,}000$ non-member sequences uniformly without replacement (seed 42) and score all 511 predicted positions per sequence with the one-epoch OLMOE target, giving $2{,}555{,}000$ tokens per corpus, all with finite, positive $\hat{\sigma}_j^2$. For binned comparisons, we group tokens into 25 bins equally spaced in $\log \hat{\sigma}_j^2$ from -8 to 2, keeping both tails in the endpoint bins. Every per-token quantity is computed before averaging. Intervals use $1{,}000$ bootstrap draws of whole sequences.

M.1 NON-MEMBER MEAN AND MEMBER SEPARATION

Non-member mean. If the target predicts domain continuations well, the gain on the observed token, G_j^{obs}, and the target-weighted mean gain μ_j should agree on average, so the centered gain $s_j = G_j^{\text{obs}} - \mu_j$ should average zero. Table 23 shows that μ_j accounts for 91.1–93.3% of the average observed non-member gain on every corpus. The remaining centered gain is small and positive, 0.025–0.034 nats. On Stanford, applying the full calibration with $\tau = 1$ at each token gives a mean calibrated gain of 0.016 (95% interval $[0.0155, 0.0167]$), the value shown in Figure 2(b).

Table 23: **The target-weighted mean gain explains most non-member gain.** Means over 5k held-out sequences per corpus (nats). The centered gain is the observed gain minus μ_j; "Explained" is the mean of μ_j divided by the mean observed gain.

Corpus	Observed gain	μ_j	Centered gain	Explained (%)
Stanford	0.429	0.400	+0.029	93.3
Khan Acad.	0.313	0.285	+0.028	91.1
Stories	0.468	0.433	+0.034	92.7
AutoMath	0.360	0.335	+0.025	93.1
Web v2	0.370	0.341	+0.028	92.3
WikiHow	0.467	0.434	+0.033	92.9

Member separation. Removing the non-member mean is useful only if members still retain a larger gain. Table 24 compares the first 5k member and 5k non-member records of each one-epoch target, using all valid predicted positions. Member means are 0.122–0.152 nats and exceed non-member means by 0.097–0.118 nats on every corpus. The non-member means, 0.025–0.035 nats, have sequence-bootstrap 95% intervals that exclude zero; they quantify the remaining departure from exact centering in Equation (11).

Table 24: **Members retain higher centered means across corpora.** Each class contains 5k sequences. Separation is the member mean minus the non-member mean.

Corpus	Centered gain (nats)		
	Mean, $M = 0$	Mean, $M = 1$	Separation
Stanford	+0.028	+0.142	+0.114
Khan Acad.	+0.029	+0.127	+0.098
Stories	+0.035	+0.152	+0.117
AutoMath	+0.025	+0.122	+0.097
Web v2	+0.029	+0.147	+0.118
WikiHow	+0.033	+0.137	+0.104

M.2 Non-member Variance and the Form of the Denominator

A near-zero average does not make individual non-member gains small. Variance weighting assumes that $\hat{\sigma}_j^2$ measures their spread at each prefix. We first state what the model predicts, then check it directly and after calibration, and finally use the results to motivate each part of the denominator.

Prediction. Under the non-member model of Proposition 1, the observed token follows the same distribution that defines μ_j and σ_j^2. The centered gain then has conditional mean zero, so its expected square equals its variance:

$$\mathbb{E}[s_j^2 \mid X_{<j}, M = 0] = \mathbb{E}[(G_j^{\text{obs}})^2 \mid X_{<j}, M = 0] - \mu_j^2 = \sigma_j^2. \tag{16}$$

Because $\tau + \sigma_j^2$ is fixed at the prefix, the calibrated gain satisfies

$$\mathbb{E}[\tilde{s}_j^2 \mid X_{<j}, M = 0] = \frac{\sigma_j^2}{(\tau + \sigma_j^2)^2}. \tag{17}$$

On a trained target we evaluate both predictions with the computed variance $\hat{\sigma}_j^2$ in place of σ_j^2, with no fitted parameters.

Direct check. Table 25 compares the measured mean of s_j^2 with the mean of $\hat{\sigma}_j^2$. Across the 25 variance bins, $\hat{\sigma}_j^2$ predicts the measured s_j^2 with token-weighted R^2 of 0.993–0.999, and over all tokens the measured value exceeds the prediction by only 2–7%. The same agreement holds after calibration: with $\tau = 1$, Equation (17) predicts the measured $\tilde{s}_j^2$ with R^2 of 0.977–0.999, and measured values exceed predictions by 0.2–9.1%. On Stanford, the paired difference between measured and predicted $\tilde{s}_j^2$ has a 95% interval of $[-0.0004, 0.0009]$, which includes zero. The computed variance therefore captures the scale of non-member fluctuations that the calibration divides by.

Table 25: **The computed variance predicts the spread of non-member centered gains.** Overall means over 5k held-out sequences per corpus. R^2 compares bin means with the unfitted predictions $\hat{\sigma}_j^2$ (Equation (16)) and $\hat{\sigma}_j^2/(1+\hat{\sigma}_j^2)^2$ (Equation (17)). The last two columns average the per-token ratio $s_j^2/\hat{\sigma}_j^2$, whose predicted value is one, below and above $\hat{\sigma}_j^2 = 0.01$.

	Overall mean		R^2 across bins		Mean $s_j^2/\hat{\sigma}_j^2$	
Corpus	$\hat{\sigma}_j^2$	s_j^2	s_j^2 vs. $\hat{\sigma}_j^2$	$\tilde{s}_j^2$ vs. pred.	$\hat{\sigma}_j^2 < 0.01$	$\hat{\sigma}_j^2 \geq 0.01$
Stanford	0.597	0.609	0.999	0.999	1.19	0.96
Khan Acad.	0.439	0.463	0.994	0.994	1.23	1.04
Stories	0.739	0.776	0.996	0.991	1.06	1.02
AutoMath	0.589	0.632	0.996	0.977	3.56	1.36
Web v2	0.584	0.617	0.993	0.993	1.49	1.02
WikiHow	0.647	0.673	0.996	0.997	2.00	0.97

Why divide by the variance. Because the spread of non-member centered gains follows $\hat{\sigma}_j^2$, the same positive centered gain is weaker evidence of membership at a prefix with large $\hat{\sigma}_j^2$. The member

signal shows the matching pattern. Pooling the full feature collections of the six corpora, with the same 25 bin edges, the member–non-member difference in mean centered gain rises with $\hat{\sigma}_j^2$ and then levels off. The saturating form $A\hat{\sigma}_j^2/(\tau_{\text{fit}} + \hat{\sigma}_j^2)$ fits the bin means with weighted $R^2 = 0.998$ ($A = 0.549$ nats, $\tau_{\text{fit}} = 1.647$ nats^2), whereas a difference proportional to $\hat{\sigma}_j^2$ fits with $R^2 = 0.829$. The saturating fit weights each bin by the inverse square of the standard error of its difference. Consider a weighted sum $\sum_j w_j s_j$ in which the weight w_j depends only on the prefix, token j has mean difference b_j between the classes, and non-member variance σ_j^2. Conditional centering removes the cross terms from the non-member variance, which becomes $\sum_j w_j^2 \sigma_j^2$. By the Cauchy–Schwarz inequality, the ratio of the squared mean difference $(\sum_j w_j b_j)^2$ to this variance is largest when $w_j \propto b_j/\sigma_j^2$. With the fitted $b_j \propto \sigma_j^2/(\tau_{\text{fit}} + \sigma_j^2)$, this gives $w_j \propto 1/(\tau_{\text{fit}} + \sigma_j^2)$, the form of DIANA's weight. A difference proportional to σ_j^2 would instead make equal weights best. This calculation motivates the shape of the weight rather than deriving it: the fit uses binned averages, and the host attacks aggregate calibrated gains in their own ways.

Why add a variance floor. The last two columns of Table 25 show where the variance estimate is least accurate. At near-deterministic prefixes with $\hat{\sigma}_j^2 < 0.01$, which contain 11–17% of tokens, the per-token ratio $s_j^2/\hat{\sigma}_j^2$ averages 1.06–3.56 instead of one: $\hat{\sigma}_j^2$ underestimates the spread there. The weight $1/\hat{\sigma}_j^2$ would amplify these tokens, consistent with the collapse of detection at $\tau = 0$ (Table 3). With $\tau > 0$, every weight is at most $1/\tau$, so these tokens contribute little after calibration. The fitted $\tau_{\text{fit}} = 1.647$ lies inside the range $\tau \in [0.25, 4]$ over which AUC stays within 0.005 of the best tested value (Appendix I), which supports the fixed choice $\tau = 1$.

N MEMBER GAINS OVER TRAINING

Proposition 2 predicts that the mean centered member gain is concave in the memorization strength, and the saturation argument in Section 3.2 predicts that it eventually falls as the memorized token dominates the mean removed by centering. This appendix describes the training ladder behind Figure 2(a), reports the calibrated gain on the same tokens, and explains how the plotted token probability relates to the model.

Checkpoint protocol. We train the OLMOE Stanford target for twenty epochs at learning rate 2×10^{-6} with three seeds, saving a checkpoint every 625 steps (one epoch). At each checkpoint, we score the same 800 member sequences per seed and average three per-token quantities over their 511 predicted positions: the target's probability of the observed token, $u_j = p_j^{\text{T}}(x_j)$; the centered gain s_j; and the calibrated gain $\tilde{s}_j = s_j/(1 + \hat{\sigma}_j^2)$, where $\hat{\sigma}_j^2$ is the variance computed from the target (Section 3.3). We verify that sequence indices, sequence lengths, and reference log-probabilities match across checkpoints, and then average the three seeds.

Centered and calibrated trajectories. Table 26 lists the checkpoint means. The token probability $\bar{u}$ increases throughout training, from 0.574 to 0.956. The mean centered gain rises from 0.121, peaks at 0.238 in epoch six, and falls to 0.089 by epoch twenty, with the peak at epoch six in every seed. The mean calibrated gain follows the same rise and fall but peaks earlier, at 0.0926 in epoch four, again in every seed. The difference comes from the variance: the mean $\hat{\sigma}_j^2$ also rises until epoch six, so dividing by $1 + \hat{\sigma}_j^2$ reduces the calibrated gains most around the peak of the centered gains. The calibrated mean remains positive at every checkpoint, ending at 0.0362, consistent with Theorem 1.

From memorization strength to token probability. Memorization strength is not observable, so Figure 2(a) uses the token probability as its horizontal axis. For one prefix and observed token x_j, write $u = p_j^+(x_j)$. Under Equation (7), $u = p_j^{\mathcal{D}}(x_j) + (1 - p_j^{\mathcal{D}}(x_j))\alpha_j$, so u increases linearly with α_j. Substituting $\alpha_j = (u - p_j^{\mathcal{D}}(x_j))/(1 - p_j^{\mathcal{D}}(x_j))$ into Equation (12) gives

$$s_j = (1-u)\left[\text{logit}(u) - \text{logit}\big(p_j^{\mathcal{D}}(x_j)\big) + \frac{\log[p_j^{\mathcal{D}}(x_j)/p_j^{\text{R}}(x_j)] - D_{\text{KL}}(p_j^{\mathcal{D}} \| p_j^{\text{R}})}{1 - p_j^{\mathcal{D}}(x_j)}\right], \tag{18}$$

where $\text{logit}(u) = \log[u/(1-u)]$. For a fixed token, the last term inside the brackets is a constant, so it contributes a term linear in u. The first term, $(1-u)\,\text{logit}(u)$, has second derivative $-1/[u^2(1-u)] <$

Table 26: **Member gains over training.** OLMoE on Stanford; means over the same 800 member sequences per seed, averaged over three seeds. The calibrated gain uses $\tau = 1$. Selected epochs of the twenty-epoch ladder.

Epoch	Token prob. $\bar{u}$	Centered gain	$\hat{\sigma}_j^2$	Calibrated gain
1	0.574	0.121	0.599	0.0670
2	0.610	0.179	0.682	0.0877
4	0.677	0.229	0.833	0.0926
6	0.746	0.238	0.889	0.0862
8	0.812	0.214	0.849	0.0743
10	0.866	0.186	0.746	0.0650
15	0.938	0.114	0.495	0.0438
20	0.956	0.089	0.401	0.0362

0 (Equation (15)). Each token's centered gain is therefore concave in its observed probability, as it is in α_j.

Fitted shape of the checkpoint means. The checkpoint means pool tokens with different domain probabilities and memorization strengths, so Equation (18) cannot be applied to them directly. We instead compare their trend with the two-parameter family $h(u) = \lambda(1-u)[\mathrm{logit}(u) - \mathrm{logit}(\pi)]$ evaluated at $u = \bar{u}$. The amplitude $\lambda > 0$ allows an overall scale, and $\pi \in (0, 1)$ is an effective baseline that absorbs the constant term. Ordinary least squares over the twenty checkpoint means, with equal weights, gives $\lambda = 0.661$, $\pi = 0.438$, and $R^2 = 0.952$; this is the dashed curve in Figure 2(a). The fit tests whether the pooled trajectory has the predicted rise and fall. Neither fitted parameter estimates a model quantity: π is not the domain probability, and $\bar{u}$ is not the mean memorization strength.

Following individual tokens. Pooling mixes tokens with different domain probabilities, which flattens the relation in Equation (18). Following each token across checkpoints removes this variation. For a fixed token, Equation (18) predicts that $s_j/(1-u_j)$ is linear in $\mathrm{logit}(u_j)$ with slope one. We test this on a separate ten-epoch run of the same recipe, using its ten checkpoints, three seeds, and 600 member sequences per seed, keeping tokens whose probability lies in $[0.02, 0.97]$. A regression with a separate intercept for each token gives slope 0.74, and the median of the per-token slopes is 0.80. Pooling different tokens without per-token intercepts gives slope 0.23. Continuing domain adaptation, which changes the effective domain probability during training, and measurement noise can both lower the within-token slope below one.

Relation to detection. The attacks saturate before the centered gain turns over. On the first five epochs of the same recipe, evaluated with $1{,}500$ target sequences per class, all three attacks exceed 99.4% TPR at 0.1% FPR by epoch four (Table 11).

O NOTATION

Table 27 collects the symbols used in the paper, grouped by role.

Table 27: **Notation used in the paper.**

Symbol	Meaning
Data and setting	
$\mathbf{x} = (x_1, \ldots, x_n)$	target text sequence; positions $j \in \{2, \ldots, n\}$
$x_{<j}$; x_j	realized prefix at position j; observed next token
$X_{<j}$, X_j	random prefix and next token; X_j is also the memorized-token draw in the member proof
$\mathcal{V}$, V	vocabulary and its size
$M \in \{0, 1\}$	fine-tuning membership indicator of the target sequence
$\mathcal{A}$	host attack's aggregation rule (mean, normalized mean, sign test)
Models and distributions	
$\mathcal{M}^{\mathrm{T}}$, $\mathcal{M}^{\mathrm{R}}$	fine-tuned target model; pre-trained reference model
p_j^{T}, p_j^{R}	their next-token distributions at prefix $x_{<j}$
ℓ^{T}, ℓ^{R}	sequence losses under target and reference (Section 2)
$\mathcal{D}$, $p_j^{\mathcal{D}}$	fine-tuning text distribution and its next-token law at $x_{<j}$
p_j^-, p_j^+	mean predictions without / with the target sequence
α_j	memorization strength; mixture weight on the memorized token (Equation (7))
Per-token quantities	
$G_j(v)$	log-gain function evaluated at token v; $(G_j(v))_{v \in \mathcal{V}}$ is the gain vector
G_j^{obs}	observed gain $G_j(x_j)$, used to form sequence-level attack scores
μ_j	target-weighted mean gain $\mathbb{E}_{p_j^{\mathrm{T}}}[G_j] = D_{\mathrm{KL}}(p_j^{\mathrm{T}} \| p_j^{\mathrm{R}})$
σ_j^2	target-weighted gain variance $\mathrm{Var}_{p_j^{\mathrm{T}}}[G_j]$
$\hat{\sigma}_j^2$	variance computed from a trained target in the empirical checks (Section 3.3)
s_j	centered gain $G_j^{\mathrm{obs}} - \mu_j$ in the membership case under consideration
$\tilde{s}_j$	calibrated gain $s_j/(\tau + \sigma_j^2)$, Equation (5)
τ	hyperparameter acting as a variance floor; $\tau = 1$ except in the τ sweep and ablation