Assessing Per-Sample Membership Inference Vulnerability without Retraining

Valentin Dorseuil *
DI ENS, École normale supérieure
Université PSL, CNRS
75005 Paris, France

Jamal Atif
CMAP, École polytechnique
Institut Polytechnique de Paris
91120 Palaiseau, France

Olivier Cappé
DI ENS, École normale supérieure
Université PSL, CNRS
75005 Paris, France

Abstract

Recent work in the privacy literature shows that sample-targeted membership inference attacks (MIAs) significantly outperform untargeted approaches by a wide margin. Motivated by this observation, we address the following question: *can the privacy vulnerability of individual training points be assessed without training shadow models?* We show that per-sample exposure to MIA is governed not only by a point's loss, but also by a data-dependent geometric measure. In the linear setting, we derive a closed-form decomposition of individual black-box MIA vulnerability into a population leverage score and a residual loss term, making explicit how sample-dependent geometry translates into privacy exposure. Since the final layer of most modern architectures is linear, we extend this framework to deep networks and propose a surrogate score operating on last-layer representations that requires only a single trained model and no shadow models. Empirical evaluations across diverse datasets and architectures show that our score outperforms loss and gradient-norm baselines at identifying the highest-risk points under state-of-the-art attacks, providing a computationally efficient and theoretically grounded tool for per-sample privacy risk assessment.

1 Introduction

Modern machine learning models, and deep neural networks in particular, are known to memorize aspects of their training data [1, 2]. This memorization induces privacy vulnerabilities that can be exploited by *Membership Inference Attacks* (MIAs), which aim to determine whether a specific data point was included in the training set or not [3, 4]. A principled defense against membership inference is provided by Differential Privacy (DP) [5], implemented in deep learning through noise injection in stochastic gradient methods [6]. However, controlling the trade-off between privacy protection and model utility remains challenging. Noise calibration typically relies on worst-case theoretical accounting, paired with empirical privacy auditing via MIAs, and often leads to either overly conservative noise levels or insufficient privacy protection. In this context, designing MIAs for empirical auditing is essential to quantify privacy leakage in non-privately trained models [7] or to validate the practical tightness of DP guarantees in privately trained models [8, 9].

*Corresponding author: valentin.dorseuil@ens.psl.eu

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

While such auditing is now standard practice [4, 8, 10], relying on aggregate metrics such as average accuracy or AUC is insufficient. These global measures can obscure critical risk heterogeneity, as outliers and rare subgroups are significantly more prone to memorization than typical samples [4, 11]. Consequently, a model certified as private on average may still expose specific points to high privacy risks. To address this heterogeneity, recent work is increasingly focusing on *per-sample* privacy risk assessment, aiming to quantify membership leakage at the level of each data point, separately, rather than in aggregate. State-of-the-art methods for per-sample auditing [4, 10] predominantly rely on *shadow models*. These techniques train multiple shadow models on random data splits to characterize each data point's influence on the model's behavior. This enables identification of which points exhibit increased sensitivity to their presence in the training dataset. However, these techniques are computationally prohibitive, especially for large-scale models, as they require retraining the model many times. This leads to our central question stated in the abstract: *can the privacy vulnerability of individual training points be assessed without training shadow models?*

To address this question in a principled way, we first consider the baseline case of Gaussian linear regression. Even in this simple model, the task poses interesting challenges: the fixed-design analysis from classical statistics leads to the *leverage score* (see below), which quantifies a data point's geometric influence solely through its feature vector, ignoring the response. The fully randomized approach, on the other hand, provides metrics that depend on general problem characteristics (dimensions, covariance operator of the features) but do not account for specific data points (see, e.g., Tan et al. [12]). To obtain relevant per-sample risk exposure metrics, we propose to consider non-asymptotic results in the randomized Gaussian setting by *conditioning on a single data point* (i.e., its feature vector and its response).

For Gaussian linear regression, we derive a closed-form expression for the expected leave-one-out generalization gap, the signal exploited by state-of-the-art black-box attacks such as LiRA [4], and show that it decomposes, at first order, into a response-free geometric term, the population leverage score $\bar{h}_{ii}$, and a response-dependent structural error ε_i. Membership inference vulnerability is therefore governed jointly by a point's geometric measure and its structural error. Samples that are both geometrically influential in shaping the model's parameters and poorly fit by the model are at the highest risk of privacy leakage.

To extend this analysis beyond the linear setting, we approximate the leave-one-out loss gap using two classical estimators: an influence-function estimate and a Newton-step estimate. Restricted to the last layer of a trained network, both admit closed-form expressions in terms of the leverage score of the learned representations, and can be computed from a single trained model at negligible cost. This yields a scalable, theoretically grounded proxy for per-sample privacy risk that matches the rankings produced by state-of-the-art shadow-model attacks, without any retraining.

The paper is organized as follows. In Section 3 we derive a closed-form expression for the expected leave-one-out generalization gap of a target point in Gaussian linear regression, showing that the signal driving LiRA-style attacks decomposes, at first order, into a label-free geometric term, the population leverage $\bar{h}_{ii}$, and a label-dependent structural error ε_i. In Section 4, we propose two retraining-free per-sample vulnerability scores for general differentiable models, based, respectively, on influence-function and Newton-step approximations of the leave-one-out loss. Restricted to the last layer, both admit closed-form expressions in terms of the leverage score of the last-layer representation, which can be efficiently computed for deep networks. Finally, Section 5 reports the performance of the proposed vulnerability scores in terms of recovery of the results obtained by state-of-the-art retraining-based MIA methods. In particular, we observe that the proposed metrics consistently outperform commonly used baselines, such as training loss or per-sample gradient-norm, across various configurations of dataset/architecture pairs. We start with a quick review of the relevant literature.

2 Related Work

Membership Inference Attacks. Membership inference aims to determine if a specific sample was used to train a model. Early approaches relied on simple metric-based classifiers, exploiting overfitting signals such as prediction confidence, entropy, or the magnitude of gradients [3, 7]. While computationally inexpensive, these methods often struggle to distinguish between a "vulnerable" member and a "hard" non-member.

To address this, current state-of-the-art methods adopt the *shadow model* paradigm [3]. By training multiple models on different data splits, attacks like LiRA (Likelihood Ratio Attack) [4] or RMIA (Robust Membership Inference Attack) [10] essentially perform a hypothesis test on the loss distribution of a target point [10]. While highly effective, these approaches are computationally prohibitive, often requiring hundreds of training runs to estimate the risk for a single point accurately. Our work seeks to achieve the precision of these hypothesis-test based methods without their computational burden, by substituting retraining with geometric analysis.

Influence Functions and Data Attribution. Recent work has begun to explore the connection between influence and privacy. Notably, Feldman and Zhang [11] and Feldman [13] link the *memorization* of a sample to its influence on the learning process, arguing that memorization is necessary for generalization in long-tailed distributions. Other work has used influence functions to craft new attack methods [14, 15], but these approaches inherit two well-known limitations of influence functions in deep learning: a prohibitive computational cost, and a fragility that often renders the resulting estimates unreliable [16].

Our work addresses these limitations by confining influence estimation to the last layer, where closed-form expressions are available. This reduces the computational overhead and avoids the fragility of full-network influence, while remaining a faithful proxy of per-sample influence as shown in our experiments.

Privacy Auditing and Heterogeneity. Differential Privacy (DP) [5] provides worst-case guarantees for membership privacy. Using algorithms such as DP-SGD [6], one can train Deep Learning models to be private. However, these bounds are often loose and do not capture the empirical reality that privacy risk is non-uniform: some outlier samples are far more exposed than others [10].

Auditing per-sample privacy risk, i.e., estimating the specific (ε, δ) DP parameters for a given sample, remains an open challenge. Existing auditing tools rely heavily on the aforementioned shadow model techniques or randomized smoothing [17], which scale poorly. Carlini et al. [4] derive per-sample vulnerability estimates from the outputs of the LiRA attacks, which need training shadow models. Song and Mittal [18] also propose a per-sample privacy risk score calibrated with shadow models. At the opposite end of the spectrum, Leemann et al. [19] empirically evaluate a large set of retraining-free predictors of per-sample membership inference success, including the baselines we compare against.

Our work contributes to this domain by proposing an efficient individual privacy-risk metric for identifying vulnerable samples, providing a first-pass, zero-retraining filter for high-risk data points. We use the LiRA-derived vulnerability of Carlini et al. [4] as our main ground truth and aim to predict it without training any shadow models. We also evaluate our metric against additional attacks [3, 10] to assess its robustness.

Removing these vulnerable points from the training set is not an option since, in practice, such deletions often expose new points, an effect described as the *privacy onion effect* by Carlini et al. [20]. Instead, our method can serve as a tool for canary selection [2] or per-sample privacy training [21, 22].

3 Analysis of Membership Vulnerability in the Linear Regression Case

To rigorously understand the privacy leakage exploited by black-box attacks, we first analyze their behavior within the context of linear models. State-of-the-art MIAs, such as shadow model classifiers [3], LiRA [4], and RMIA [10], compute membership scores via classifiers or likelihood ratio tests fitted to per-sample losses. In practice, they estimate the loss distribution of each sample, conditioned on whether it was included in the training set, by training shadow models on multiple data splits.

To model the stochasticity of training on random subsamples, we isolate a fixed target point i and treat the remainder of the dataset as being drawn from an underlying probability distribution. We then analyze the marginal distribution of the target's loss under two scenarios: the IN setting, where the model is trained on the fixed point alongside the samples drawn from the distribution, and the OUT setting, where the model is trained exclusively on the sampled distribution without the target point.

3.1 Problem Setup and Notation

Let $\mathcal{Z} \subset \mathbb{R}^p \times \mathbb{R}$ be the data space, and let $z_i = (x_i, y_i) \in \mathcal{Z}$ be a fixed target observation. Throughout our analysis, we condition on this target point, treating it as a deterministic constant. We consider a full training dataset S of size n that contains this fixed target z_i. The other $n-1$ points in S, denoted collectively as $S_{-i} = \{(X_j, Y_j)\}_{j \neq i}$, are random variables sampled from the underlying data distribution.

Because z_i is fixed, the randomness in our estimators arises entirely from these $n-1$ background observations. We assume their covariates are drawn i.i.d. from a multivariate Gaussian distribution, $X_j \sim \mathcal{N}(0, \Sigma)$, and their responses follow the linear generative model $Y_j \mid X_j \sim \mathcal{N}(X_j^\top \theta, \sigma^2)$ where $\theta \in \mathbb{R}^p$ is the unknown true parameter and σ^2 the structural noise variance. Equivalently, $Y_{-i} = X_{-i}\theta + \varepsilon_{-i}$, where $\varepsilon_{-i} \in \mathbb{R}^{n-1}$ is the noise vector, with $\varepsilon_{-i} \sim \mathcal{N}(0, \sigma^2 I_{n-1})$ independent of X_{-i}. We consider the scalar response case for simplicity, and the results are extended to multivariate responses in Appendix A.

Let $X \in \mathbb{R}^{n \times p}$ denote the full design matrix, where the i-th row is the deterministic $x_i^\top$ and the remaining rows form the random submatrix $X_{-i} \in \mathbb{R}^{(n-1)\times p}$. Similarly, let $Y \in \mathbb{R}^n$ be the response vector containing y_i and the random subvector $Y_{-i} \in \mathbb{R}^{n-1}$.

Let $\hat{\theta} = (X^\top X)^{-1} X^\top Y$ be the ordinary least squares (OLS) estimator trained on the full dataset S, and $\hat{\theta}_{-i}$ be the estimator trained on S_{-i}. We denote $e_i = y_i - x_i^\top \hat{\theta}$ the residual for the target point under the full model and $e_{-i} = y_i - x_i^\top \hat{\theta}_{-i}$ the residual for the target point under the leave-one-out model. The corresponding squared residuals are the training loss $\ell_i = (y_i - x_i^\top \hat{\theta})^2$ and the leave-one-out loss $\ell_{-i} = (y_i - x_i^\top \hat{\theta}_{-i})^2$.

We introduce the leverage matrix $H = X(X^\top X)^{-1} X^\top$, and denote $h_{ii} = x_i^\top (X^\top X)^{-1} x_i$ the empirical leverage score of the target point, which is the i-th diagonal element of H. This score is a random variable due to its dependence on X_{-i}, and quantifies the geometric influence of the point on its own fitted value [23] illustrated by the leave-one-out identity $e_i = (1 - h_{ii})\, e_{-i}$ [24].

We define two deterministic quantities characterizing the target point: its population leverage score $\bar{h}_{ii} = x_i^\top \Sigma^{-1} x_i$, which corresponds to the squared Mahalanobis distance to the feature distribution, and its structural error $\varepsilon_i = y_i - x_i^\top \theta$.

Our goal is to characterize the distribution of the losses ℓ_i and ℓ_{-i} and see how these distributions relate to the target point's MIA vulnerability.

3.2 Finite-Sample Distributions of Leverage Score and Loss

We begin by analyzing the leave-one-out Gram matrix $G_{-i} = X_{-i}^\top X_{-i}$, which follows a Wishart distribution $\mathcal{W}_p(\Sigma, n-1)$ [25]. Applying the Sherman–Morrison–Woodbury identity allows us to express both the empirical leverage score h_{ii} and the leave-one-out residual e_{-i} as functions of G_{-i}, thereby isolating the geometric randomness of the design from the structural noise. The following lemma makes this decomposition explicit.

Lemma 3.1 (Joint Law of Leverage Scores and Residuals)**.** *Let $G_{-i} = X_{-i}^\top X_{-i}$ denote the leave-one-out Gram matrix, which follows a Wishart distribution $\mathcal{W}_p(\Sigma, n-1)$ [25]. We define*

$$V = \frac{\bar{h}_{ii}}{x_i^\top G_{-i}^{-1} x_i}, \qquad Z = -\frac{\sqrt{V}}{\sigma\sqrt{\bar{h}_{ii}}}\, x_i^\top G_{-i}^{-1} X_{-i}^\top \varepsilon_{-i}. \tag{1}$$

Then, the empirical leverage score h_{ii} and the leave-one-out residual e_{-i} satisfy the identities:

$$h_{ii} = \frac{\bar{h}_{ii}}{V + \bar{h}_{ii}}, \qquad e_{-i} = \varepsilon_i + \sigma \frac{\sqrt{\bar{h}_{ii}}}{\sqrt{V}} Z. \tag{2}$$

Moreover, conditioned on (x_i, y_i), $V \sim \chi^2_{n-p}$ and $Z \sim \mathcal{N}(0,1)$, with $Z \perp\!\!\!\perp V$.

The independence of Z and V in Lemma 3.1 is an important property: given X_{-i}, the standardized prediction error Z is $\mathcal{N}(0,1)$ regardless of the design, hence independent of any function of it, including V. Combining the leave-one-out identity $e_i = (1 - h_{ii}) e_{-i}$ [24], which gives $\ell_i =$

$(1-h_{ii})^2\ell_{-i}$, with Lemma 3.1, we obtain a full characterization of the joint distribution of the losses at point i in Proposition 3.2.

Proposition 3.2 (Per-sample Loss Distribution)**.** *Under the Gaussian linear regression model, the leave-one-out loss ℓ_{-i} and the training loss ℓ_i at a point (x_i, y_i) satisfy*

$$\ell_{-i} = \left(\varepsilon_i + \sigma\frac{\sqrt{\bar{h}_{ii}}}{\sqrt{V}}Z\right)^2, \qquad \ell_i = \left(\varepsilon_i + \sigma\frac{\sqrt{\bar{h}_{ii}}}{\sqrt{V}}Z\right)^2\left(\frac{V}{V+\bar{h}_{ii}}\right)^2, \tag{3}$$

where V and Z are defined in Lemma 3.1, satisfying $V \sim \chi^2_{n-p}$ and $Z \sim \mathcal{N}(0,1)$, with $Z \perp\!\!\!\perp V$.

3.3 Per-Point Vulnerability under Optimal Inference

Using Proposition 3.2 and the Neyman–Pearson Lemma [26], we derive the optimal black-box attack in closed form. In the large-sample regime $n-p \gg 1$, the variable $V \sim \chi^2_{n-p}$ concentrates sharply around its mean, so we replace it by the constant $n-p$ and Proposition 3.2 reduces to a deterministic rescaling in distribution: $\ell_i \mid \mathcal{H}_{\text{IN}} = c_i^2 \cdot (\ell_{-i} \mid \mathcal{H}_{\text{OUT}})$, with the scale factor $c_i^2 = ((n-p)/(n-p+\bar{h}_{ii}))^2 < 1$. The likelihood ratio is then monotone in the observed loss, so the Neyman–Pearson optimal test thresholds $\ell_{i,\text{obs}}$ directly.

Corollary 3.3 (Optimal MIA Trade-off)**.** *Let $F_{i,OUT}$ denote the CDF of ℓ_{-i} under $\mathcal{H}_{OUT}$. For any target false positive rate $\alpha_i \in (0,1)$, the false negative rate of the Neyman–Pearson optimal attack at point i is*

$$\beta_i(\alpha_i) \;=\; 1 - F_{i,OUT}\Big(\tfrac{1}{c_i^2}\,F^{-1}_{i,OUT}(\alpha_i)\Big)\,. \tag{4}$$

The closed-form expression itself follows from a standard Neyman–Pearson argument; the corollary's value lies in clarifying *which* features of the problem dictate vulnerability. The full trade-off function $\alpha_i \mapsto \beta_i(\alpha_i)$ depends on the data only through $(\bar{h}_{ii}, \varepsilon_i)$ and the global parameters (n, p, σ^2). In particular, the entire design matrix X_{-i} enters only through the scalar population leverage $\bar{h}_{ii}$. The pair $(\bar{h}_{ii}, \varepsilon_i)$ thus acts as a per-point sufficient statistic for membership-inference vulnerability.

3.4 The Expected Generalization Gap

Alongside the trade-off curve, the expected generalization gap $\mathbb{E}[\Delta\ell_i]$ has been identified as the fundamental theoretical bound on the power of loss-based MIAs [7]. In our model, it admits a theoretical closed-form and a tractable asymptotic expression that make the dependence on $(\bar{h}_{ii}, \varepsilon_i)$ explicit.

Proposition 3.4 (Expected Generalization Gap)**.** *For $n-p > 4$, the expected difference between the leave-one-out and training loss for a fixed point (x_i, y_i) is exactly:*

$$\mathbb{E}[\Delta\ell_i] = \mathbb{E}_{V\sim\chi^2_{n-p}}\left[\left(\varepsilon_i^2 + \sigma^2\frac{\bar{h}_{ii}}{V}\right)\frac{\bar{h}_{ii}(2V+\bar{h}_{ii})}{(V+\bar{h}_{ii})^2}\right]. \tag{5}$$

The asymptotic expansion of this gap in powers of $(n-p)^{-1}$ is:

$$\mathbb{E}[\Delta\ell_i] = \frac{2\varepsilon_i^2\bar{h}_{ii}}{n-p} + \frac{4\varepsilon_i^2\bar{h}_{ii} + \bar{h}_{ii}^2(2\sigma^2 - 3\varepsilon_i^2)}{(n-p)^2} + \mathcal{O}\left(\frac{1}{(n-p)^3}\right). \tag{6}$$

Proposition 3.4 establishes that, to first order, the generalization gap is governed by the interaction between a point's structural error ε_i^2 and its leverage $\bar{h}_{ii}$. Intuitively, the points exhibiting the largest generalization gap are those that are simultaneously outliers in feature space (high $\bar{h}_{ii}$) and difficult to predict (high ε_i^2). Moreover, the gap vanishes at a rate of $1/n$, which confirms that the privacy risk associated with any individual point diminishes as the size of the dataset grows.

Figure 1 illustrates Corollary 3.3 and Proposition 3.4 using Monte-Carlo simulations. The left panel shows how the expected generalization gap grows with the true residual ε_i^2 and the leverage score $\bar{h}_{ii}$. The middle and right panels present the trade-off curves of the optimal MIA for varying ε_i and $\bar{h}_{ii}$, respectively, confirming that the attack's statistical power increases monotonically in both parameters.

Proofs for all the results presented in this section are deferred to Appendix A.

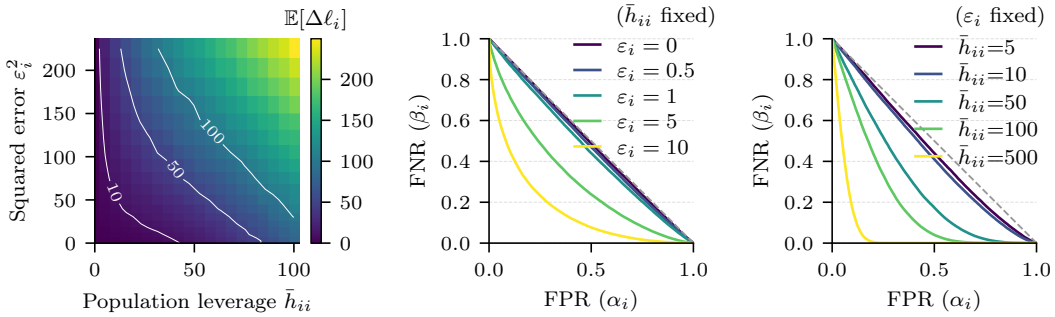

Figure 1: Generalization gap and membership inference attack performance for the Gaussian linear regression model. **(left)** Heatmap of the generalization gap $\mathbb{E}[\Delta\ell_i]$ as a function of ε_i^2 and $\bar{h}_{ii}$, estimated with Monte-Carlo. **(middle)** Theoretical false negative rate (FNR) β_i of the optimal attack on the point i as a function of the false positive rate (FPR) α_i, for different values of the structural error ε_i, with $\bar{h}_{ii} = 1$ and $\sigma^2 = 1$. **(right)** Same as (middle) but with fixed $\varepsilon_i = 1$ and $\bar{h}_{ii}$ varying.

4 Generalization to Non-Linear Models

In the linear regression case, we could derive the exact value of the loss gap $\ell_{-i} - \ell_i$ without retraining. This is because the OLS estimator has a closed-form solution and the loss is quadratic. For more general models, such as generalized linear models or deep neural networks, we do not have closed-form solutions. In the following, we discuss how common techniques from influence functions and Newton steps can be used to estimate the loss gap without retraining, and how they relate to the previous analysis in both the linear regression and the logistic regression case.

4.1 Influence Functions and Newton Step

Influence Function. Influence functions [27, 28] provide a way to estimate the effect of removing a training point z_i on the model parameters $\hat{\theta}$ without retraining. Formally, the influence of z_i is defined as:

$$I(z_i) = -\Big[\frac{1}{n}\sum_{j=1}^{n}\nabla_\theta^2\ell(\hat{\theta}, z_j)\Big]^{-1}\nabla_\theta\ell(\hat{\theta}, z_i).$$

Using a first-order Taylor expansion, this defines our first estimator for the loss gap, which we call the *Influence Function estimator*:

$$\widehat{\Delta\ell}_i^{\mathrm{IF}} = \nabla_\theta\ell(\hat{\theta}, z_i)^\top\Big[\sum_{j=1}^{n}\nabla_\theta^2\ell(\hat{\theta}, z_j)\Big]^{-1}\nabla_\theta\ell(\hat{\theta}, z_i). \tag{7}$$

Newton Step. Alternatively, one can approximate the loss gap by taking a single Newton step on the leave-one-out objective starting from $\hat{\theta}$. Using a first-order Taylor expansion on the loss, this defines our second estimator, which we call the *Newton Step estimator*:

$$\widehat{\Delta\ell}_i^{\mathrm{NS}} = -\nabla_\theta\ell(\hat{\theta}, z_i)^\top\Big[\frac{1}{n}\sum_{j\neq i}\nabla_\theta^2\ell(\hat{\theta}, z_j)\Big]^{-1}\frac{1}{n}\sum_{j\neq i}\nabla_\theta\ell(\hat{\theta}, z_j).$$

Notably, when the model has fully converged and minimizes the empirical risk (i.e., $\frac{1}{n}\sum_{j=1}^{n}\nabla_\theta\ell(\hat{\theta}, z_j) = 0$), it follows that $\frac{1}{n}\sum_{j\neq i}\nabla_\theta\ell(\hat{\theta}, z_j) = -\frac{1}{n}\nabla_\theta\ell(\hat{\theta}, z_i)$. Under this convergence assumption, the Newton Step estimator simplifies to:

$$\widehat{\Delta\ell}_i^{\mathrm{NS}} = \nabla_\theta\ell(\hat{\theta}, z_i)^\top\Big[\sum_{j\neq i}\nabla_\theta^2\ell(\hat{\theta}, z_j)\Big]^{-1}\nabla_\theta\ell(\hat{\theta}, z_i), \tag{8}$$

a form also known as the *Approximate Leave-One-Out* (ALO) estimator in the statistics literature [29] and used in approximate unlearning methods [30, 31].

Remarkably, $\widehat{\Delta\ell}_i^{\mathrm{IF}}$ and $\widehat{\Delta\ell}_i^{\mathrm{NS}}$ coincide up to a rank-one correction of the Hessian, despite being derived from distinct principles. In the following section, we derive closed-form expressions for these two estimators in the context of linear and logistic regression, and show how they can serve as scalable proxies for the loss gap to evaluate vulnerability to membership inference attacks.

4.2 Analysis for Linear Regression and Logistic Regression

We now derive the exact expressions of $\widehat{\Delta\ell}_i^{\mathrm{IF}}$ and $\widehat{\Delta\ell}_i^{\mathrm{NS}}$ for linear and logistic regression. These are not further approximations: they are the direct, closed-form instantiations of Equations (7) and (8) for the respective model classes.

Linear Regression. In linear regression, the OLS estimator $\hat{\theta} = (X^\top X)^{-1} X^\top Y$ is available in closed form and the loss is quadratic, so both estimators admit exact closed-form expressions. Letting e_i denote the residual and h_{ii} the leverage score, the two estimators yield:

$$\widehat{\Delta\ell}_i^{\mathrm{IF}} = 2e_i^2 h_{ii}, \qquad \widehat{\Delta\ell}_i^{\mathrm{NS}} = \frac{2e_i^2 h_{ii}}{1 - h_{ii}}. \tag{9}$$

The Newton Step estimator introduces a leverage correction $(1 - h_{ii})^{-1}$, bringing it closer to the exact leave-one-out formula: $\Delta\ell_i = e_i^2(2h_{ii} - h_{ii}^2)/(1 - h_{ii})^2$, particularly for high-leverage points.

Logistic Regression. In binary logistic regression, with $\hat{p}_i = s(\hat{\theta}^\top x_i)$, the two estimators yield:

$$\widehat{\Delta\ell}_i^{\mathrm{IF}} = \frac{(y_i - \hat{p}_i)^2}{w_i} h_{ii}, \qquad \widehat{\Delta\ell}_i^{\mathrm{NS}} = \frac{(y_i - \hat{p}_i)^2}{w_i} \frac{h_{ii}}{1 - h_{ii}}, \tag{10}$$

where $w_i = \hat{p}_i(1 - \hat{p}_i)$, s is the sigmoid function and $h_{ii} = w_i x_i^\top (X^\top W X)^{-1} x_i$ is the weighted leverage score, with $W = \mathrm{diag}(w_j)_{j=1,\dots,n}$. This is the logistic regression analogue of the OLS leverage score [32]: in both cases, h_{ii} captures the local sensitivity of the fitted response to the target, $\partial\hat{y}_i/\partial y_i$ in linear regression and $\partial\hat{p}_i/\partial y_i$ in logistic regression. As in Equation (9), the Newton Step estimator applies a leverage correction relative to the Influence Function estimator.

We generalize both of these results for the multivariate case in Appendix B.6. Motivated by these findings, as well as our theoretical analysis in Proposition 3.4, we propose to use these estimates as proxies for vulnerability assessment in deep neural networks. In practice, we consider only the linear last layer of the network, and apply the estimators defined in equations (9) and (10) (or their multivariate counterparts) depending on the loss used for the training (cross-entropy for classification, quadratic for regression).

We emphasize that these estimates are designed to evaluate a point's inherent vulnerability (its outlier status within the dataset distribution, by approximating the loss gap) and not its vulnerability to one specific model instance[2].

4.3 Differences Between the Two Approaches

In both linear and logistic regression, the two estimators differ by a single multiplicative factor: $\widehat{\Delta\ell}_i^{\mathrm{NS}} / \widehat{\Delta\ell}_i^{\mathrm{IF}} = 1/(1 - h_{ii})$. At low leverage they therefore agree to leading order. The correction matters only as $h_{ii} \to 1$, a regime reached, for instance, when x_i is orthogonal to every other row of X, in which case $h_{ii} = 1$ exactly. We have:

$$\frac{\widehat{\Delta\ell}_i^{\mathrm{IF}}}{\Delta\ell_i} \underset{h_{ii}\to 1}{\sim} 2(1 - h_{ii})^2, \qquad \frac{\widehat{\Delta\ell}_i^{\mathrm{NS}}}{\Delta\ell_i} \underset{h_{ii}\to 1}{\sim} 2(1 - h_{ii}).$$

As $h_{ii} \to 1$, both estimators vanish relative to $\Delta\ell_i$, but at different rates: the Influence Function estimator captures only an $O((1 - h_{ii})^2)$ fraction of the true loss change, whereas the Newton Step estimator captures an $O(1 - h_{ii})$ fraction. The Newton Step therefore tracks the per-sample generalization gap substantially better than the Influence Function at high-leverage points.

Proofs for all the results presented in this section are deferred to Appendix B.

[2] A point's vulnerability is defined as how strongly it leaks across training randomness. Evaluating this vulnerability with respect to a single model instance alone is ill-defined, as one (point, model) pair yields a single attack outcome.

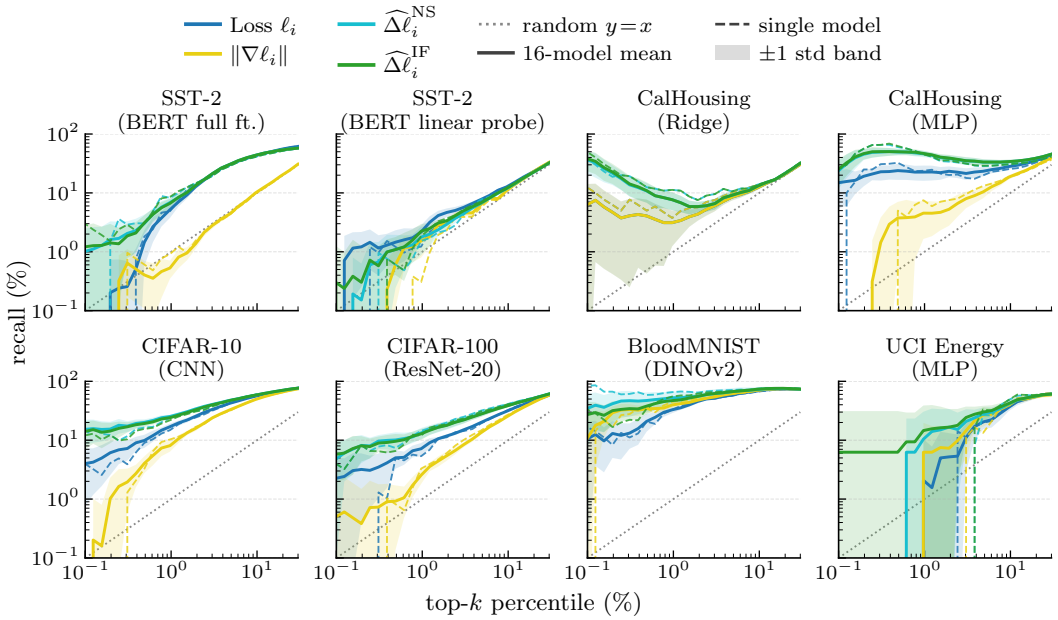

Figure 2: Recall of the $k\%$ of points most vulnerable to LiRA (ranked by the per-sample Attack Success Rate) when keeping the top-$k\%$ for each of the four surrogates: the loss ℓ_i, the gradient norm $\|\nabla \ell_i\|$, the influence estimate $\widehat{\Delta\ell}_i^{\mathrm{IF}}$, and the Newton estimate $\widehat{\Delta\ell}_i^{\mathrm{NS}}$. Axes are log-log; the dotted diagonal $y = x$ is the random baseline. Solid lines are means over 16 target models with ± 1 std bands; dashed-lines show a single representative model.

5 Experiments

We evaluate our two surrogates $\widehat{\Delta\ell}_i^{\mathrm{IF}}$ and $\widehat{\Delta\ell}_i^{\mathrm{NS}}$ for per-sample membership-inference vulnerability across a diverse set of modalities, architectures and learning regimes.

5.1 Experimental Setup

Datasets and architectures. We cover both classification and regression on tabular, vision and text data: (i) *CIFAR-10* [33] with a small CNN; (ii) *CIFAR-100* [33] with ResNet-20 [34]; (iii) *BloodMNIST* [35–37] with a DINOv2 ViT-small classifier [38], trained with a linear probe on top of frozen features; (iv) *SST-2* [39] with a BERT [40] classifier in two regimes: full fine-tuning of `bert_uncased_L-2_H-128_A-2` [41] and a linear probe on top of a frozen `bert-base-uncased`; (v) *California Housing* [42] with both a ridge regression and a deep MLP; (vi) *UCI Energy* [43] with a tabular MLP. In Appendix D.4 we also conduct experiments on a larger dataset, with a variant as we use the K-Fac [44] approximation for the computation of the inverse-Hessian-vector products. For each task we use the standard train split, randomly partitioned in half between members and non-members ($N_{\text{train}} = N/2$) with a per-model seed, following the LiRA shadow-model protocol [4].

Shadow and target models. For every dataset/architecture pair we train 200 shadow (reference) models with i.i.d. random member/non-member splits, plus 16 independent target models used to compute the per-sample vulnerability surrogates. Optimizers, learning rates, batch sizes, weight decay and number of epochs follow standard recipes for each architecture (AdamW for vision and text, SGD-momentum for ResNet-20, Adam for the regression MLPs); the exact hyper-parameters are listed in Appendix C.1 and in the release scripts.

Attacks and surrogates. We use LiRA [4] as the reference per-sample attack, since it achieves the highest Attack Success Rate (ASR) in our experiments. As our ground truth, we choose the per-sample ASR computed over the 200 reference models, rather than the TPR@low-FPR. As a

Table 1: Recall (%) of the true LiRA top-1 % when keeping the top-5 % by each surrogate. Mean ± std across target models. Higher is better; **bold** = best per dataset.

		Recall of Top 1% @5% (in %)			
Dataset	Model	Loss ℓ_i	$\|\nabla \ell_i\|$	$\widehat{\Delta\ell}_i^{\mathrm{NS}}$	$\widehat{\Delta\ell}_i^{\mathrm{IF}}$
SST-2	BERT, full ft.	$28.2 \pm 2.4\,\%$	$5.3 \pm 1.1\,\%$	$28.8 \pm 2.6\,\%$	$\mathbf{28.8 \pm 2.6}\,\%$
	BERT, linear probe	$\mathbf{10.7 \pm 1.6}\,\%$	$7.2 \pm 1.9\,\%$	$8.2 \pm 2.5\,\%$	$8.9 \pm 1.6\,\%$
CalHousing	Ridge regression	$12.3 \pm 2.8\,\%$	$12.3 \pm 2.8\,\%$	$\mathbf{13.3 \pm 2.6}\,\%$	$13.2 \pm 2.5\,\%$
	MLP	$52.2 \pm 6.7\,\%$	$22.5 \pm 6.9\,\%$	$74.9 \pm 7.4\,\%$	$\mathbf{75.5 \pm 7.0}\,\%$
CIFAR-10	CNN	$53.7 \pm 2.5\,\%$	$36.8 \pm 2.5\,\%$	$\mathbf{62.8 \pm 2.1}\,\%$	$62.2 \pm 2.2\,\%$
CIFAR-100	ResNet-20	$23.5 \pm 1.5\,\%$	$12.3 \pm 1.8\,\%$	$\mathbf{36.8 \pm 5.1}\,\%$	$35.2 \pm 4.6\,\%$
BloodMNIST	DINOv2	$74.3 \pm 6.4\,\%$	$82.3 \pm 3.9\,\%$	$\mathbf{90.4 \pm 12.7}\,\%$	$87.1 \pm 9.0\,\%$
UCI Energy	MLP	$12.5 \pm 16.7\,\%$	$14.6 \pm 21.0\,\%$	$33.3 \pm 34.4\,\%$	$\mathbf{39.6 \pm 32.7}\,\%$

per-sample metric, the latter depends on the extreme tails of the shadow model loss distribution, and is therefore noisier than the per-sample ASR[3].

We compare four *retraining-free* surrogates of vulnerability, all computed from a *single* target model and *without* shadow models: the training loss ℓ_i, the gradient-norm $\|\nabla \ell_i\|$, the influence-function estimate $\widehat{\Delta\ell}_i^{\mathrm{IF}}$ and the Newton-step estimate $\widehat{\Delta\ell}_i^{\mathrm{NS}}$. All experiments were performed on points in the training set (*members*). The same quantities can be computed from the loss observed on *non-members* by reversing the equations; see Appendix D.3 for details.

For deep networks, both variants are restricted to the last layer, which keeps their cost essentially negligible. In Appendix D, we extend this analysis to include comparisons with both the RMIA attack [10] and the original shadow-model attack [3]. For two settings, CIFAR-10 with a CNN and California Housing with an MLP, we additionally computed the full-network influence function estimate of the loss, without the restriction to the last layer, i.e., $\nabla_\theta \ell_i^\top H_{\hat{\theta}}^{-1} \nabla_\theta \ell_i$. We compute it using the inverse- -Hessian-vector Products [45] and the Conjugate Gradient algorithm [46]. More details can be found in Appendix D.1.

Metrics. Following standard per-sample auditing practice [3, 4, 10], we evaluate surrogates based on top-k% recall: the fraction of the most vulnerable points (highest ASR) correctly identified within the surrogate's top-k% rankings. The Spearman rank correlation is provided in Appendix D.5 for reference, but we find it less informative in our context as it is dominated by the high density of points near the 50% ASR (random guess), where surrogates provide little discriminative signal.

All experiments were run on an NVIDIA V100 GPU cluster. The full pipeline (training, statistics, attacks, scores and figures) is reproducible from the released code available at `https://github.com/vdorseuil/per_sample_membership`.

5.2 Results

We present the recall @5% of the true LiRA top-1% (measured by per-sample Attack Success Rate) in Table 1. Figure 2 illustrates the performance of these surrogates for multiple recall percentages[4]. We also present more extensive results for different percentages of recall in Appendix D, which show similar trends.

Across seven out of eight settings, our two surrogates $\widehat{\Delta\ell}_i^{\mathrm{IF}}$ and $\widehat{\Delta\ell}_i^{\mathrm{NS}}$, consistently recover the samples LiRA flags as most vulnerable, well above the random baseline, often by one to two orders

[3]This conclusion applies specifically to estimating the vulnerability of individual samples. As an aggregate metric, the TPR@low-FPR is often preferred, as it is more informative about the leakage of the most vulnerable points [4]. For completeness, we also compared with the TPR@low-FPR and provide the results in Appendix D.2.

[4]At very small top-k percentiles, the sample size of considered points can be extremely low. Consequently, the random baseline represents a very small expected value. Due to variance across just 16 random seeds, it is highly probable to observe a lower recovery rate than the mathematical expectation. This explains the drop of the recall (y-axis) for low top-k percentiles (x-axis).

of magnitude. The results are more mixed on BERT with linear probing. We suspect this arises from the fact that the model learned very little during the training process, as the training loss did not decrease significantly.

They also surpass the loss baseline on seven out of eight settings, with recall@1% improving by more than 20% on some datasets. The sole exception is the BERT linear probe, where the loss baseline marginally outperforms our method; we hypothesize that this setting, characterized by simple training dynamics and few vulnerable points, is one in which the raw loss already provides a strong signal. We further observe high variance in recall@1% across all surrogates on the UCI Energy dataset, which we attribute to the mismatch between the small dataset size ($N = 768$) and the high-dimensional tabular MLP ($p = 512$). At this scale, recall@1% is evaluated over only 7 points, making the metric inherently noisy.

5.3 Discussion

Across settings, our leverage-based surrogates have greater difficulty recovering the points most vulnerable to RMIA and classifier-based attacks than those vulnerable to LiRA, specifically in the regression case (Appendix D.6). This follows from our construction: in the linear model, our surrogate is built as an unbiased estimator of the loss gap, measuring the difference in loss when the target point is included in versus excluded from the dataset. This quantity is precisely the fundamental signal governing the LiRA attack, which approximates the likelihood ratio of the loss under these two hypotheses. In contrast, the RMIA attack estimates a pairwise likelihood ratio that captures how the loss changes when the target point is replaced by another one, rather than removed. Likewise, the classifier-based approach learns a non-parametric decision boundary that might not be well captured by the leverage-based scores. This contrast explains the stronger recovery of points most vulnerable to LiRA. Despite these differences, our method still outperforms the loss baseline on RMIA in all cases, with results nearly matching those on LiRA in the classification case. Finally, the full-network influence function proved worse in both speed and recall compared to our last-layer approach (see Appendix D.1).

Overall, these scores yield a shadow-free, single-model proxy that approaches LiRA's per-point ASR ranking at a fraction of its computational cost. They can be used by model developers or privacy auditors to detect the most vulnerable points easily. Scoring all points with our leverage-based surrogates on a single target model is $1{,}000\times$ to $10{,}000\times$ faster than training the 200 shadow models required by LiRA, excluding the target-model training cost, which is shared by both approaches. Detailed timings for each surrogate and attack are reported in the Appendix C.2.

6 Conclusion

We proposed a retraining-free framework for per-sample membership inference vulnerability assessment. In Gaussian linear regression, we derived a closed-form decomposition of the expected leave-one-out gap into a response-free population leverage term and a response-dependent structural error. We extended this view to deep networks via influence-function and Newton-step surrogates restricted to the last layer, both computable from a single trained model. Across seven out of eight dataset/architecture pairs, our surrogates outperform loss and gradient-norm baselines at recovering LiRA's most vulnerable points, at a fraction of the cost.

Limitations remain: the theory is exact only in the linear case, the last-layer restriction overlooks earlier feature learning, and the surrogates track LiRA more closely than RMIA or classifier-based attacks. Natural extensions include layer-wise or kernel-based leverage scores, integration into DP-SGD training as a canary-selection or per-sample weighting tool, and a finer characterization of which memorization signals escape local quadratic approximations.

Acknowledgements

This work is part of the project SSF-ML-DH of the PEPR Digital Health and was supported by the French National Research Agency, as a part of the France 2030 program, under grant ANR-22-PESN-0014-01. Furthermore, we thank GENCI at IDRIS for providing computer and storage resources through the grant 2026-AD011017108 on the supercomputer Jean Zay.

We are also grateful to Hanna Benarroch for her helpful feedback on the project.

References

[1] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

[2] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, pages 267–284, 2019.

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

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

[5] Cynthia Dwork. Differential privacy. In *International Colloquium on Automata, Languages and Programming (ICALP)*, pages 1–12, 2006.

[6] Martín Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 308–318, 2016.

[7] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *IEEE Computer Security Foundations Symposium (CSF)*, pages 268–282, 2018.

[8] Milad Nasr, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlini. Adversary instantiation: Lower bounds for differentially private machine learning. In *IEEE Symposium on Security and Privacy (SP)*, pages 866–882, 2021.

[9] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private SGD? In *Advances in Neural Information Processing Systems*, pages 22205–22216, 2020.

[10] Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks. In *International Conference on Machine Learning*, pages 58244–58282, 2024.

[11] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In *Advances in Neural Information Processing Systems*, pages 2881–2891, 2020.

[12] Jasper Tan, Blake Mason, Hamid Javadi, and Richard Baraniuk. Parameters or privacy: A provable tradeoff between overparameterization and membership inference. In *Advances in Neural Information Processing Systems*, pages 17488–17500, 2022.

[13] Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 954–959, 2020.

[14] Gilad Cohen and Raja Giryes. Membership inference attack using self influence functions. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4880–4889, 2024.

[15] Anshuman Suri, Xiao Zhang, and David Evans. Do parameters reveal more than loss for membership inference? *Transactions on Machine Learning Research*, 2024.

[16] Samyadeep Basu, Phillip Pope, and Soheil Feizi. Influence functions in deep learning are fragile. In *International Conference on Learning Representations*, 2021.

[17] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *IEEE Symposium on Security and Privacy (SP)*, pages 656–672, 2019.

[18] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *USENIX Security Symposium*, pages 2615–2632, 2021.

[19] Tobias Leemann, Bardh Prenkaj, and Gjergji Kasneci. Is my data safe? predicting instance-level membership inference success for white-box and black-box attacks. In *ICML 2024 Next Generation of AI Safety Workshop*, 2024.

[20] Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramèr. The privacy onion effect: Memorization is relative. In *Advances in Neural Information Processing Systems*, pages 13263–13276, 2022.

[21] Yu-Xiang Wang. Per-instance differential privacy. *Journal of Privacy and Confidentiality*, 9(1), 2019.

[22] Mohammad Alaggan, Sébastien Gambs, and Anne-Marie Kermarrec. Heterogeneous differential privacy. *Journal of Privacy and Confidentiality*, 7(2), 2017.

[23] David A. Belsley, Edwin Kuh, and Roy E. Welsch. Detecting influential observations and outliers. In *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*, chapter 2, pages 6–84. John Wiley & Sons, 1980.

[24] David M. Allen. The relationship between variable selection and data agumentation and a method for prediction. *Technometrics*, 16(1):125–127, 1974.

[25] Robb J. Muirhead. The multivariate linear model. In *Aspects of Multivariate Statistical Theory*, chapter 10, pages 429–525. John Wiley & Sons, 1982.

[26] Jerzy Neyman and Egon Sharpe Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A*, 231:289–337, 1933.

[27] Frank R. Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, 1974.

[28] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894, 2017.

[29] Kamiar Rahnama Rad and Arian Maleki. A scalable estimate of the out-of-sample prediction error via approximate leave-one-out cross-validation. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 82(4):965–996, 2020.

[30] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *International Conference on Machine Learning*, pages 3832–3842, 2020.

[31] Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *Advances in Neural Information Processing Systems*, pages 18075–18086, 2021.

[32] Daryl Pregibon. Logistic regression diagnostics. *The Annals of Statistics*, 9(4):705–724, 1981.

[33] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[35] Andrea Acevedo, Anna Merino, Santiago Alférez, Ángel Molina, Laura Boldú, and José Rodellar. A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. *Data in Brief*, 30:105474, 2020.

[36] Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 191–195, 2021.

[37] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

[38] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[39] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, 2013.

[40] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 4171–4186, 2019.

[41] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models. arXiv:1908.08962, 2019.

[42] R. Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.

[43] Athanasios Tsanas and Angeliki Xifara. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings*, 49: 560–567, 2012.

[44] James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning*, pages 2408–2417, 2015.

[45] Barak A. Pearlmutter. Fast exact multiplication by the hessian. *Neural Computation*, 6(1): 147–160, 1994.

[46] Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

[47] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[48] 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 *Network and Distributed System Security Symposium (NDSS)*, 2019.

[49] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Table of Contents

A Omitted Proofs from Section 3 **14**

A.1 Proof of Lemma 3.1 (Joint Law of Leverage Scores and Residuals) 14

A.2 Proof of Proposition 3.2 (Per-sample Loss Distribution) 15

A.3 Proof of Corollary 3.3 (Optimal MIA Trade-off) 16

A.4 Proof of Proposition 3.4 (Expected Generalization Gap) 16

A.5 Generalization to the Multivariate Case 17

B Omitted Proofs from Section 4 **18**

B.1 Proof of the Influence Function Estimate (Eq. (7)) 18

B.2 Proof of the Newton Step Estimate (Eq. (8)) 19

B.3 Results in the Linear Regression (Eq. (9)) 19

B.4 Results in the Logistic Regression (Eq. (10)) 20

B.5 Proof that $\partial\hat{p}_i/\partial y_i = h_{ii}$ 20

B.6 Generalization to the Multivariate Case 21

C Details on the Experimental Setup **23**

C.1 Model Architectures and Training Hyperparameters 23

C.2 Time and Compute Resources 24

D Additional Experimental Results **24**

D.1 Comparison with Full-Network Influence functions 25

D.2 Per-Sample TPR@low-FPR as Baseline 25

D.3 Evaluation on Non-Members 25

D.4 Evaluation on a Larger Dataset 26

D.5 Top-k Recall and Spearman Correlation for LiRA Attack 26

D.6 Top-k Recall and Spearman Correlation for RMIA Attack and Classifier Shokri's Attack 26

D.7 Additional Visualizations 28

A Omitted Proofs from Section 3

This appendix provides the rigorous proofs for the theoretical results characterizing the finite-sample generalization gap. Throughout these proofs, we maintain the conditioning on the fixed target observation (x_i, y_i), treating x_i and y_i as deterministic constants. The randomness originates entirely from X_{-i} and Y_{-i}.

A.1 Proof of Lemma 3.1 (Joint Law of Leverage Scores and Residuals)

Proof. We first derive the distribution of the empirical leverage score h_{ii}. With $G_{-i} = X_{-i}^\top X_{-i}$, the full Gram matrix can be decomposed as $X^\top X = G_{-i} + x_i x_i^\top$. The empirical leverage score for the i-th observation is $h_{ii} = x_i^\top (X^\top X)^{-1} x_i$.

Applying the Sherman-Morrison formula to invert the rank-one update:

$$(G_{-i} + x_i x_i^\top)^{-1} = G_{-i}^{-1} - \frac{G_{-i}^{-1} x_i x_i^\top G_{-i}^{-1}}{1 + x_i^\top G_{-i}^{-1} x_i}$$

Multiplying both sides by $x_i^\top$ on the left and x_i on the right yields:

$$h_{ii} = x_i^\top G_{-i}^{-1} x_i - \frac{(x_i^\top G_{-i}^{-1} x_i)^2}{1 + x_i^\top G_{-i}^{-1} x_i}$$

Let $\tilde{h}_{ii} = x_i^\top G_{-i}^{-1} x_i$. Simplifying the fraction, we obtain $h_{ii} = \frac{\tilde{h}_{ii}}{1+\tilde{h}_{ii}}$.

Because the rows of X_{-i} are i.i.d. $\mathcal{N}(0, \Sigma)$, G_{-i} follows a Wishart distribution $\mathcal{W}_p(\Sigma, n-1)$. A standard property of the Inverse-Wishart distribution states that for any fixed vector x_i, the quadratic form scales as an inverse chi-squared random variable. Specifically:

$$\frac{x_i^\top \Sigma^{-1} x_i}{x_i^\top G_{-i}^{-1} x_i} \sim \chi^2_{(n-1)-p+1} = \chi^2_{n-p}$$

Let this random variable be denoted as V. Recalling that $\bar{h}_{ii} = x_i^\top \Sigma^{-1} x_i$, we have $V = \bar{h}_{ii}/\tilde{h}_{ii}$, which implies $\tilde{h}_{ii} = \bar{h}_{ii}/V$. Substituting this into our expression for h_{ii} gives:

$$h_{ii} = \frac{\bar{h}_{ii}/V}{1 + \bar{h}_{ii}/V} = \frac{\bar{h}_{ii}}{V + \bar{h}_{ii}}$$

which concludes the first part of the lemma.

We now derive the distribution of the leave-one-out residual e_{-i}. The leave-one-out OLS estimator is $\hat{\theta}_{-i} = G_{-i}^{-1} X_{-i}^\top Y_{-i}$. Under the true model, $Y_{-i} = X_{-i}\theta + \varepsilon_{-i}$, where $\varepsilon_{-i} \sim \mathcal{N}(0, \sigma^2 I_{n-1})$ is unconditionally independent of X_{-i}. Substituting this yields $\hat{\theta}_{-i} = \theta + G_{-i}^{-1} X_{-i}^\top \varepsilon_{-i}$.

The leave-one-out residual is:

$$e_{-i} = y_i - x_i^\top \hat{\theta}_{-i} = (y_i - x_i^\top \theta) - x_i^\top G_{-i}^{-1} X_{-i}^\top \varepsilon_{-i} = \varepsilon_i - U$$

where $\varepsilon_i = y_i - x_i^\top \theta$ is a deterministic constant and $U = x_i^\top G_{-i}^{-1} X_{-i}^\top \varepsilon_{-i}$.

We now condition on X_{-i}. Since $\varepsilon_{-i} \perp\!\!\!\perp X_{-i}$, the conditional distribution of ε_{-i} remains $\mathcal{N}(0, \sigma^2 I_{n-1})$. Because U is a linear combination of a Gaussian vector, $U \mid X_{-i}$ is Gaussian. Its conditional moments are:

$$\mathbb{E}[U \mid X_{-i}] = x_i^\top G_{-i}^{-1} X_{-i}^\top \mathbb{E}[\varepsilon_{-i}] = 0 \tag{11}$$

$$\text{Var}(U \mid X_{-i}) = x_i^\top G_{-i}^{-1} X_{-i}^\top (\sigma^2 I_{n-1}) X_{-i} G_{-i}^{-1} x_i = \sigma^2 x_i^\top G_{-i}^{-1} G_{-i} G_{-i}^{-1} x_i = \sigma^2 \tilde{h}_{ii} \tag{12}$$

Substituting $\tilde{h}_{ii} = \bar{h}_{ii}/V$, we have $U \mid X_{-i} \sim \mathcal{N}(0, \sigma^2 \bar{h}_{ii}/V)$. Let $Z = -U/(\sigma\sqrt{\bar{h}_{ii}}/\sqrt{V})$. Then $Z \mid X_{-i} \sim \mathcal{N}(0, 1)$.

Crucially, because the conditional distribution of Z given X_{-i} is standard normal and does not depend on X_{-i}, Z is unconditionally independent of X_{-i}. Since V is entirely a function of X_{-i}, it strictly follows that $Z \perp\!\!\!\perp V$. Thus, we can write the residual as $e_{-i} = \varepsilon_i + \sigma \frac{\sqrt{\bar{h}_{ii}}}{\sqrt{V}} Z$.

Finally, by standard regression update rules, the training residual is $e_i = e_{-i}/(1 + \tilde{h}_{ii})$. Squaring this yields the training loss:

$$\ell_i = e_i^2 = \left(\frac{e_{-i}}{1 + \bar{h}_{ii}/V}\right)^2 = e_{-i}^2 \left(\frac{V}{V + \bar{h}_{ii}}\right)^2$$

□

A.2 Proof of Proposition 3.2 (Per-sample Loss Distribution)

Proof. The results follow directly from Lemma 3.1 by writing that the training loss is

$$\ell_i = e_i^2 = e_{-i}^2 (1 - h_{ii})^2$$

and the leave-one-out loss is $\ell_{-i} = e_{-i}^2$.

□

A.3 Proof of Corollary 3.3 (Optimal MIA Trade-off)

Proof. Under the non-member hypothesis $\mathcal{H}_{\text{OUT}}$, the optimal decision rule rejects the null hypothesis if the observed loss is unusually small: $\ell_{\text{obs},i} < \gamma_\alpha$. For a fixed False Positive Rate α, the threshold is strictly determined by the non-member cumulative distribution function: $\alpha = \mathbb{P}(\ell_{\text{obs},i} < \gamma_\alpha \mid \mathcal{H}_{\text{OUT}})$, yielding $\gamma_\alpha = F^{-1}_{i,\text{OUT}}(\alpha)$.

Under the member hypothesis $\mathcal{H}_{\text{IN}}$, the loss is scaled by c_i^2. The True Positive Rate is the probability that this scaled loss falls below the same threshold:

$$\text{TPR}(\alpha) = \mathbb{P}\left(c_i^2 \ell_{\text{obs},i} < \gamma_\alpha \mid \mathcal{H}_{\text{OUT}}\right) = \mathbb{P}\left(\ell_{\text{obs},i} < \frac{\gamma_\alpha}{c_i^2} \mid \mathcal{H}_{\text{OUT}}\right) = F_{i,\text{OUT}}\left(\frac{\gamma_\alpha}{c_i^2}\right)$$

We finish the proof by writing $\beta_i(\alpha) = \text{FNR}(\alpha) = 1 - \text{TPR}(\alpha)$ □

A.4 Proof of Proposition 3.4 (Expected Generalization Gap)

Proof. The generalization gap is $\Delta\ell_i = \ell_{-i} - \ell_i$. Using the relationship $\ell_i = e_{-i}^2[V/(V + \bar{h}_{ii})]^2$, we can factor out e_{-i}^2:

$$\Delta\ell_i = e_{-i}^2\left(1 - \frac{V^2}{(V+\bar{h}_{ii})^2}\right) = e_{-i}^2\left(\frac{(V+\bar{h}_{ii})^2 - V^2}{(V+\bar{h}_{ii})^2}\right) = e_{-i}^2\frac{\bar{h}_{ii}(2V+\bar{h}_{ii})}{(V+\bar{h}_{ii})^2}$$

To compute the expectation $\mathbb{E}[\Delta\ell_i \mid x_i, y_i]$, we use the independence of Z and V established in Lemma 3.1. We first take the inner expectation with respect to Z:

$$\mathbb{E}_Z[e_{-i}^2 \mid V] = \mathbb{E}_Z\left[\left(\varepsilon_i + \sigma\frac{\sqrt{\bar{h}_{ii}}}{\sqrt{V}}Z\right)^2\right] \tag{13}$$

$$= \varepsilon_i^2 + 2\varepsilon_i\sigma\frac{\sqrt{\bar{h}_{ii}}}{\sqrt{V}}\mathbb{E}[Z] + \sigma^2\frac{\bar{h}_{ii}}{V}\mathbb{E}[Z^2] = \varepsilon_i^2 + \sigma^2\frac{\bar{h}_{ii}}{V} \tag{14}$$

Substituting this inner expectation into the gap yields the exact finite-sample formula:

$$\mathbb{E}[\Delta\ell_i \mid x_i, y_i] = \mathbb{E}_{V\sim\chi^2_{n-p}}\left[\left(\varepsilon_i^2 + \sigma^2\frac{\bar{h}_{ii}}{V}\right)\frac{\bar{h}_{ii}(2V+\bar{h}_{ii})}{(V+\bar{h}_{ii})^2}\right]$$

To derive the asymptotic expansion, let $k = n - p$ and define $x = 1/V$. We expand the rational term as $x \to 0$ using the Maclaurin series for $(1 + \bar{h}_{ii}x)^{-2}$:

$$1 - (1+\bar{h}_{ii}x)^{-2} = 1 - (1 - 2\bar{h}_{ii}x + 3\bar{h}_{ii}^2x^2 - \dots) = 2\bar{h}_{ii}x - 3\bar{h}_{ii}^2x^2 + \mathcal{O}(x^3)$$

Multiplying by the expanded residual term $\mathbb{E}_Z[e_{-i}^2] = \varepsilon_i^2 + \sigma^2\bar{h}_{ii}x$:

$$\begin{aligned} g(x) &= (\varepsilon_i^2 + \sigma^2\bar{h}_{ii}x)(2\bar{h}_{ii}x - 3\bar{h}_{ii}^2x^2 + \mathcal{O}(x^3)) \\ &= 2\varepsilon_i^2\bar{h}_{ii}x + \bar{h}_{ii}^2(2\sigma^2 - 3\varepsilon_i^2)x^2 + \mathcal{O}(x^3) \end{aligned} \tag{15}$$

We now apply the exact inverse moments of the chi-squared distribution $V \sim \chi^2_k$. For $k > 4$:

$$\mathbb{E}[x] = \mathbb{E}[V^{-1}] = \frac{1}{k-2}, \quad \mathbb{E}[x^2] = \mathbb{E}[V^{-2}] = \frac{1}{(k-2)(k-4)}$$

Substituting these moments into our expansion:

$$\mathbb{E}[\Delta\ell_i] = \frac{2\varepsilon_i^2\bar{h}_{ii}}{k-2} + \frac{\bar{h}_{ii}^2(2\sigma^2 - 3\varepsilon_i^2)}{(k-2)(k-4)} + \mathcal{O}\left(k^{-3}\right)$$

Finally, we convert this falling-factorial series into standard powers of $1/k$ using geometric series expansions: $(k-2)^{-1} = k^{-1} + 2k^{-2} + \mathcal{O}(k^{-3})$ and $(k-2)^{-1}(k-4)^{-1} = k^{-2} + \mathcal{O}(k^{-3})$. Gathering like terms gives:

$$\begin{aligned} \mathbb{E}[\Delta\ell_i] &= 2\varepsilon_i^2\bar{h}_{ii}\left(\frac{1}{k} + \frac{2}{k^2}\right) + \bar{h}_{ii}^2(2\sigma^2 - 3\varepsilon_i^2)\left(\frac{1}{k^2}\right) + \mathcal{O}\left(\frac{1}{k^3}\right) \\ &= \frac{2\varepsilon_i^2\bar{h}_{ii}}{k} + \frac{4\varepsilon_i^2\bar{h}_{ii} + \bar{h}_{ii}^2(2\sigma^2 - 3\varepsilon_i^2)}{k^2} + \mathcal{O}\left(\frac{1}{k^3}\right) \end{aligned} \tag{16}$$

Substituting $k = n - p$ completes the proof. □

A.5 Generalization to the Multivariate Case

We now extend Proposition 3.4 to a q-dimensional response. Let $Y_j \in \mathbb{R}^q$ be generated as $Y_j \mid X_j \sim \mathcal{N}(\Theta^\top X_j, \Sigma_y)$ with $\Theta \in \mathbb{R}^{p\times q}$ and $\Sigma_y \succ 0$. We collect responses in $Y \in \mathbb{R}^{n\times q}$, fit the OLS estimator $\hat{\Theta} = (X^\top X)^{-1} X^\top Y$, and define the training and leave-one-out losses

$$\ell_i = \|y_i - \hat{\Theta}^\top x_i\|^2, \qquad \ell_{-i} = \|y_i - \hat{\Theta}_{-i}^\top x_i\|^2.$$

The structural error is now a vector $\varepsilon_i = y_i - \Theta^\top x_i \in \mathbb{R}^q$.

Proposition A.1 (Multivariate generalization gap)**.** *For $n - p > 4$, conditional on (x_i, y_i),*

$$\mathbb{E}[\Delta\ell_i \mid x_i, y_i] = \mathbb{E}_{V\sim\chi^2_{n-p}}\left[\left(\|\varepsilon_i\|^2 + \operatorname{tr}(\Sigma_y)\,\frac{\bar{h}_{ii}}{V}\right)\frac{\bar{h}_{ii}(2V+\bar{h}_{ii})}{(V+\bar{h}_{ii})^2}\right], \tag{17}$$

with the asymptotic expansion in $k = n - p$

$$\mathbb{E}[\Delta\ell_i \mid x_i, y_i] = \frac{2\|\varepsilon_i\|^2\bar{h}_{ii}}{n-p} + \frac{4\|\varepsilon_i\|^2\bar{h}_{ii} + \bar{h}_{ii}^2\big(2\operatorname{tr}(\Sigma_y) - 3\|\varepsilon_i\|^2\big)}{(n-p)^2} + \mathcal{O}\!\left(\frac{1}{(n-p)^3}\right). \tag{18}$$

Proof. The leverage analysis of Lemma 3.1 depends only on X and is unchanged: $h_{ii} = \bar{h}_{ii}/(V+\bar{h}_{ii})$ with $V \sim \chi^2_{n-p}$ and $\bar{h}_{ii} = x_i^\top \Sigma^{-1} x_i$.

Writing $E = Y - X\Theta \in \mathbb{R}^{n\times q}$ for the noise matrix (rows i.i.d. $\mathcal{N}(0, \Sigma_y)$), we have $\hat{\Theta}_{-i} = \Theta + G_{-i}^{-1} X_{-i}^\top E_{-i}$, hence

$$e_{-i} = y_i - \hat{\Theta}_{-i}^\top x_i = \varepsilon_i - U, \qquad U = E_{-i}^\top X_{-i} G_{-i}^{-1} x_i \in \mathbb{R}^q.$$

Conditional on X_{-i}, U is a linear combination of the independent rows of E_{-i}, hence Gaussian with $\mathbb{E}[U \mid X_{-i}] = 0$ and

$$\operatorname{Cov}(U \mid X_{-i}) = \big(x_i^\top G_{-i}^{-1} X_{-i}^\top X_{-i} G_{-i}^{-1} x_i\big)\,\Sigma_y = \tilde{h}_{ii}\,\Sigma_y = \frac{\bar{h}_{ii}}{V}\,\Sigma_y,$$

using $\tilde{h}_{ii} = \bar{h}_{ii}/V$ from the proof of Lemma 3.1. Setting $Z = -\sqrt{V}/\sqrt{\bar{h}_{ii}} \cdot U$, we obtain $Z \mid X_{-i} \sim \mathcal{N}(0, \Sigma_y)$. Since this conditional law does not depend on X_{-i}, Z is unconditionally independent of X_{-i} and therefore of V. We thus write

$$e_{-i} = \varepsilon_i + \frac{\sqrt{\bar{h}_{ii}}}{\sqrt{V}}\,Z, \qquad Z \sim \mathcal{N}(0, \Sigma_y), \quad V \sim \chi^2_{n-p}, \quad Z \perp\!\!\!\perp V.$$

The univariate Sherman-Morrison contraction $e_i = (1 - h_{ii})\,e_{-i}$ applies column-wise to $\hat{\Theta}$, so it carries over to vector residuals and yields $\ell_i = (1 - h_{ii})^2\,\ell_{-i}$.

Combining the two displays,

$$\Delta\ell_i = \ell_{-i}\big(1 - (1-h_{ii})^2\big) = \|e_{-i}\|^2 \cdot \frac{\bar{h}_{ii}(2V+\bar{h}_{ii})}{(V+\bar{h}_{ii})^2}.$$

By independence of Z and V,

$$\mathbb{E}_Z\big[\|e_{-i}\|^2 \,\big|\, V\big] = \|\varepsilon_i\|^2 + 2\,\frac{\sqrt{\bar{h}_{ii}}}{\sqrt{V}}\,\varepsilon_i^\top \mathbb{E}[Z] + \frac{\bar{h}_{ii}}{V}\,\mathbb{E}[Z^\top Z] = \|\varepsilon_i\|^2 + \operatorname{tr}(\Sigma_y)\,\frac{\bar{h}_{ii}}{V},$$

which yields the exact finite-sample expression in the proposition. The asymptotic expansion follows the same algebra as the proof of Proposition 3.4, replacing ε_i^2 by $\|\varepsilon_i\|^2$ and σ^2 by $\operatorname{tr}(\Sigma_y)$.

□

The univariate result of Proposition 3.4 is recovered when $q = 1$ and $\Sigma_y = \sigma^2$. Crucially, the leverage $\bar{h}_{ii}$ remains label-free and so retains the same first-order role; only the structural-noise contribution is replaced by the sum of marginal output variances $\operatorname{tr}(\Sigma_y)$.

B Omitted Proofs from Section 4

B.1 Proof of the Influence Function Estimate (Eq. (7))

We recall here the derivation of the influence function estimate, following the perturbation-based approach popularized by Koh and Liang [28].

Proof. Let the empirical risk over the full dataset be $R(\theta) = \frac{1}{n}\sum_{j=1}^{n} \ell(\theta, z_j)$, and let $\hat{\theta} = \arg\min_\theta R(\theta)$ be the empirical risk minimizer.

Rather than strictly removing the i-th point, we consider the effect of infinitesimally up-weighting the loss of z_i by some small continuous value ϵ. We define the perturbed empirical risk minimizer as:

$$\hat{\theta}_{\epsilon,z_i} \triangleq \arg\min_\theta \left(\frac{1}{n}\sum_{j=1}^{n} \ell(\theta, z_j) + \epsilon \ell(\theta, z_i) \right)$$

By the first-order optimality condition, the gradient of the perturbed objective at $\hat{\theta}_{\epsilon,z_i}$ must be zero:

$$0 = \frac{1}{n}\sum_{j=1}^{n} \nabla_\theta \ell(\hat{\theta}_{\epsilon,z_i}, z_j) + \epsilon \nabla_\theta \ell(\hat{\theta}_{\epsilon,z_i}, z_i)$$

We then differentiate this entire condition with respect to ϵ and evaluate it at $\epsilon = 0$ (where $\hat{\theta}_{\epsilon,z_i} = \hat{\theta}$). Applying the chain rule yields:

$$0 = \left[\frac{1}{n}\sum_{j=1}^{n} \nabla_\theta^2 \ell(\hat{\theta}, z_j) \right] \left. \frac{d\hat{\theta}_{\epsilon,z_i}}{d\epsilon} \right|_{\epsilon=0} + \nabla_\theta \ell(\hat{\theta}, z_i)$$

Let $H = \frac{1}{n}\sum_{j=1}^{n} \nabla_\theta^2 \ell(\hat{\theta}, z_j)$ be the Hessian of the empirical risk. Solving for the derivative gives us the influence of up-weighting z_i on the parameters:

$$\mathcal{I}_{\text{up,params}}(z_i) \triangleq \left. \frac{d\hat{\theta}_{\epsilon,z_i}}{d\epsilon} \right|_{\epsilon=0} = -H^{-1} \nabla_\theta \ell(\hat{\theta}, z_i)$$

Removing the point z_i from the dataset corresponds to reducing its weight from $\frac{1}{n}$ to 0. We can approximate this by setting $\epsilon = -\frac{1}{n}$. Using a first-order Taylor expansion around $\epsilon = 0$, the change in the parameters is:

$$\hat{\theta}_{-i} - \hat{\theta} \approx \left. \frac{d\hat{\theta}_{\epsilon,z_i}}{d\epsilon} \right|_{\epsilon=0} \cdot \left(-\frac{1}{n} \right) = \frac{1}{n} H^{-1} \nabla_\theta \ell(\hat{\theta}, z_i)$$

Finally, to estimate the loss gap for the i-th point, we apply a first-order Taylor expansion to the loss function itself:

$$\ell_{-i} - \ell_i \approx \nabla_\theta \ell(\hat{\theta}, z_i)^\top (\hat{\theta}_{-i} - \hat{\theta}) \approx \frac{1}{n} \nabla_\theta \ell(\hat{\theta}, z_i)^\top H^{-1} \nabla_\theta \ell(\hat{\theta}, z_i).$$

Expanding the Hessian H and pulling the factor n out of the inverse gives the final influence function estimate:

$$\Delta\ell_i \approx \nabla_\theta \ell(\hat{\theta}, z_i)^\top \Big[\sum_{j=1}^{n} \nabla_\theta^2 \ell(\hat{\theta}, z_j) \Big]^{-1} \nabla_\theta \ell(\hat{\theta}, z_i)$$

This gives us our influence function estimate for the loss gap $\widehat{\Delta\ell}_i^{\text{IF}}$ when removing the i-th point from the training set. □

B.2 Proof of the Newton Step Estimate (Eq. (8))

Proof. Let $\hat{\theta}$ be the empirical risk minimizer and define the leave-one-out objective

$$\mathcal{L}_{-i}(\theta) = \frac{1}{n} \sum_{j \neq i} \ell(\theta, z_j).$$

Write $g_{-i} = \nabla_\theta \mathcal{L}_{-i}(\hat{\theta}) = \frac{1}{n} \sum_{j \neq i} \nabla_\theta \ell(\hat{\theta}, z_j)$ and $H_{-i} = \nabla^2_\theta \mathcal{L}_{-i}(\hat{\theta}) = \frac{1}{n} \sum_{j \neq i} \nabla^2_\theta \ell(\hat{\theta}, z_j)$, and assume H_{-i} is invertible.

The leave-one-out minimizer $\hat{\theta}_{-i} = \arg\min_\theta \mathcal{L}_{-i}(\theta)$ is approximated by *one Newton step* on $\mathcal{L}_{-i}$ started at $\hat{\theta}$:

$$\hat{\theta}_{-i} - \hat{\theta} \;\approx\; -H_{-i}^{-1} \, g_{-i}.$$

The loss gap is $\Delta\ell_i = \ell(\hat{\theta}_{-i}, z_i) - \ell(\hat{\theta}, z_i)$. A first-order Taylor expansion of $\ell(\cdot, z_i)$ around $\hat{\theta}$ gives

$$\Delta\ell_i \;\approx\; \nabla_\theta \ell(\hat{\theta}, z_i)^\top (\hat{\theta}_{-i} - \hat{\theta}) \;\approx\; -\nabla_\theta \ell(\hat{\theta}, z_i)^\top H_{-i}^{-1} \, g_{-i}.$$

This is the Newton Step estimator

$$\widehat{\Delta\ell}_i^{\mathrm{NS}} = -\nabla_\theta \ell(\hat{\theta}, z_i)^\top \Big[\frac{1}{n} \sum_{j \neq i} \nabla^2_\theta \ell(\hat{\theta}, z_j) \Big]^{-1} \frac{1}{n} \sum_{j \neq i} \nabla_\theta \ell(\hat{\theta}, z_j).$$

When the model has fully converged and minimizes the empirical risk, it follows that $\frac{1}{n} \sum_{j \neq i} \nabla_\theta \ell(\hat{\theta}, z_j) = -\frac{1}{n} \nabla_\theta \ell(\hat{\theta}, z_i)$, hence

$$g_{-i} = \frac{1}{n} \sum_{j \neq i} \nabla_\theta \ell(\hat{\theta}, z_j) = -\frac{1}{n} \nabla_\theta \ell(\hat{\theta}, z_i).$$

Substituting, gives

$$\widehat{\Delta\ell}_i^{\mathrm{NS}} = \nabla_\theta \ell(\hat{\theta}, z_i)^\top \Big[\sum_{j \neq i} \nabla^2_\theta \ell(\hat{\theta}, z_j) \Big]^{-1} \nabla_\theta \ell(\hat{\theta}, z_i),$$

recovering Eq. (8). □

This estimate differs from the influence function only through the leave-one-out Hessian, which accounts for the removal of z_i's own curvature (a rank-one correction), and is exact for quadratic losses up to the first-order expansion of $\ell(\cdot, z_i)$.

B.3 Results in the Linear Regression (Eq. (9))

We specialize Eq. (7) and Eq. (8) to the squared loss $\ell(\theta, z_j) = (y_j - x_j^\top \theta)^2$ on the dataset $\{(x_j, y_j)\}_{j=1}^n$. Throughout, $e_i = y_i - x_i^\top \hat{\theta}$ denotes the training residual and $h_{ii} = x_i^\top (X^\top X)^{-1} x_i$ the leverage score.

Proof. For each j,

$$\nabla_\theta \ell(\theta, z_j) = -2(y_j - x_j^\top \theta) \, x_j, \qquad \nabla^2_\theta \ell(\theta, z_j) = 2 \, x_j x_j^\top.$$

Evaluated at $\hat{\theta}$, $\nabla_\theta \ell(\hat{\theta}, z_i) = -2 e_i \, x_i$.

Influence. The full-data Hessian is $\sum_{j=1}^n \nabla^2_\theta \ell(\hat{\theta}, z_j) = 2 \, X^\top X$. Substituting in Eq. (7):

$$\widehat{\Delta\ell}_i^{\mathrm{IF}} = (-2 e_i x_i)^\top (2 \, X^\top X)^{-1} (-2 e_i x_i) = 2 \, e_i^2 \, x_i^\top (X^\top X)^{-1} x_i = 2 \, e_i^2 \, h_{ii}.$$

Newton. The leave-one-out Hessian is $\sum_{j\neq i} \nabla^2_\theta \ell(\hat{\theta}, z_j) = 2\,(X^\top X - x_i x_i^\top) = 2\,G_{-i}$. Sherman-Morrison gives

$$x_i^\top G_{-i}^{-1} x_i = \frac{x_i^\top (X^\top X)^{-1} x_i}{1 - x_i^\top (X^\top X)^{-1} x_i} = \frac{h_{ii}}{1 - h_{ii}}.$$

Substituting in Eq. (8):

$$\widehat{\Delta\ell}_i^{\mathrm{NS}} = (-2e_i x_i)^\top (2\,G_{-i})^{-1} (-2e_i x_i) = 2\,e_i^2 \cdot \frac{h_{ii}}{1 - h_{ii}}.$$

The Newton estimate is exact up to the first-order Taylor expansion of the loss; for the quadratic OLS loss, the second-order remainder is itself $e_i^2\, h_{ii}^2/(1-h_{ii})^2$, recovering the exact leave-one-out loss gap. □

B.4 Results in the Logistic Regression (Eq. (10))

We specialize Eq. (7) and Eq. (8) to the binary cross-entropy loss with $\hat{p}_j = s(x_j^\top \hat{\theta})$ and $w_j = \hat{p}_j(1-\hat{p}_j)$. Let $W = \mathrm{diag}(w_j)_{j=1}^n$ and define the leverage score for the logistic model [32] as:

$$h_{ii} = w_i\, x_i^\top (X^\top W X)^{-1} x_i.$$

Proof. The per-sample loss $\ell(\theta, z_j) = -y_j \log \hat{p}_j - (1-y_j)\log(1-\hat{p}_j)$ has gradient and Hessian

$$\nabla_\theta \ell(\theta, z_j) = -(y_j - \hat{p}_j)\, x_j, \qquad \nabla^2_\theta \ell(\theta, z_j) = w_j\, x_j x_j^\top,$$

so the empirical Hessian is $\sum_{j=1}^n \nabla^2_\theta \ell(\hat{\theta}, z_j) = X^\top W X$.

Influence. Substituting in Eq. (7):

$$\widehat{\Delta\ell}_i^{\mathrm{IF}} = (y_i - \hat{p}_i)^2\, x_i^\top (X^\top W X)^{-1} x_i = \frac{(y_i - \hat{p}_i)^2}{w_i}\, h_{ii}.$$

Newton. The leave-one-out Hessian is $X^\top W X - w_i x_i x_i^\top$. By Sherman-Morrison,

$$x_i^\top \left(X^\top W X - w_i x_i x_i^\top\right)^{-1} x_i = \frac{x_i^\top (X^\top W X)^{-1} x_i}{1 - w_i\, x_i^\top (X^\top W X)^{-1} x_i} = \frac{h_{ii}/w_i}{1 - h_{ii}}.$$

Substituting in Eq. (8):

$$\widehat{\Delta\ell}_i^{\mathrm{NS}} = (y_i - \hat{p}_i)^2 \cdot \frac{h_{ii}/w_i}{1 - h_{ii}} = \frac{(y_i - \hat{p}_i)^2}{w_i} \cdot \frac{h_{ii}}{1 - h_{ii}}.$$

□

B.5 Proof that $\partial \hat{p}_i / \partial y_i = h_{ii}$

Proof. The MLE $\hat{\theta}$ satisfies the score equation

$$X^\top (y - \hat{p}) = 0,$$

where $\hat{p} = s(X\hat{\theta})$. Differentiating with respect to y_i and applying the chain rule, with $\partial \hat{p}/\partial \hat{\eta} = W$ and $\hat{\eta} = X\hat{\theta}$, yields

$$x_i \;=\; X^\top W X\, \frac{\partial \hat{\theta}}{\partial y_i}, \qquad \text{hence} \qquad \frac{\partial \hat{\theta}}{\partial y_i} \;=\; (X^\top W X)^{-1} x_i.$$

Since $\partial \hat{p}_i / \partial y_i = w_i\, \partial \hat{\eta}_i / \partial y_i = w_i\, x_i^\top (\partial \hat{\theta}/\partial y_i)$, we obtain

$$\frac{\partial \hat{p}_i}{\partial y_i} \;=\; w_i\, x_i^\top (X^\top W X)^{-1} x_i \;=\; h_{ii}.$$

□

B.6 Generalization to the Multivariate Case

We extend the closed-form expressions to multivariate-output settings: multivariate linear regression and multinomial logistic regression. The arguments rely only on the per-sample gradient and Hessian, so Eq. (7) and Eq. (8) apply. We derive the exact formulas for the influence and Newton estimates in these two cases.

Multivariate linear regression. For multivariate linear regression over q outputs with quadratic loss $\ell(\Theta) = \sum_{j=1}^{n} \|y_j - \Theta^\top x_j\|^2$, the problem decouples into q *independent* OLS regressions. For a single dimension k, the residual is $e_{i,k} \in \mathbb{R}$, the gradient is $g_{i,k} = -2e_{i,k}x_i$, and the scalar leverage is $h_{ii} = x_i^\top (X^\top X)^{-1} x_i$.

Proposition B.1 (Influence Approximation)**.** *The influence approximation (using the full-data Hessian) for the leave-one-out change in loss is:*

$$\widehat{\Delta\ell}_i^{\mathrm{IF}} = 2\|e_i\|^2 h_{ii} \tag{19}$$

where $e_i = y_i - \hat{\Theta}^\top x_i \in \mathbb{R}^q$ is the full residual vector.

Proof. The full Hessian for dimension k is $\mathcal{H}_\Theta = 2X^\top X$ and is independent of k. The loss gap for dimension k using the influence approximation presented in equation (7) is:

$$\widehat{\Delta\ell}_{ik}^{\mathrm{IF}} = g_{i,k}^\top \mathcal{H}_\Theta^{-1} g_{i,k} = (-2e_{i,k}x_i)^\top (2X^\top X)^{-1}(-2e_{i,k}x_i) = 2e_{i,k}^2 x_i^\top (X^\top X)^{-1} x_i = 2e_{i,k}^2 h_{ii}$$

Summing the independent changes over all q dimensions yields the total loss gap:

$$\widehat{\Delta\ell}_i^{\mathrm{IF}} = \sum_{k=1}^{q} 2e_{i,k}^2 h_{ii} = 2h_{ii} \sum_{k=1}^{q} e_{i,k}^2 = 2\|e_i\|^2 h_{ii}$$

□

Proposition B.2 (Newton Approximation)**.** *The Newton approximation, using the leave-one-out Hessian for the leave-one-out change in loss is:*

$$\widehat{\Delta\ell}_i^{\mathrm{NS}} = 2\|e_i\|^2 \frac{h_{ii}}{1 - h_{ii}} \tag{20}$$

Proof. The exact LOO Hessian for dimension k is $H_{-i} = 2(X^\top X - x_i x_i^\top)$. Applying the Sherman-Morrison formula to invert this rank-1 downdate yields $x_i^\top (X^\top X - x_i x_i^\top)^{-1} x_i = \frac{h_{ii}}{1-h_{ii}}$. The loss gap per dimension using this LOO Hessian is:

$$\widehat{\Delta\ell}_{ik}^{\mathrm{NS}} = g_{i,k}^\top H_{-i}^{-1} g_{i,k} = (-2e_{i,k}x_i)^\top \left[2(X^\top X - x_i x_i^\top)\right]^{-1} (-2e_{i,k}x_i)$$

$$\widehat{\Delta\ell}_{ik}^{\mathrm{NS}} = 4e_{i,k}^2 \cdot \frac{1}{2} x_i^\top (X^\top X - x_i x_i^\top)^{-1} x_i = 2e_{i,k}^2 \frac{h_{ii}}{1 - h_{ii}}$$

Summing over all q dimensions yields the total loss gap:

$$\widehat{\Delta\ell}_i^{\mathrm{NS}} = \sum_{k=1}^{q} 2e_{i,k}^2 \frac{h_{ii}}{1 - h_{ii}} = 2\frac{h_{ii}}{1 - h_{ii}} \sum_{k=1}^{q} e_{i,k}^2 = 2\|e_i\|^2 \frac{h_{ii}}{1 - h_{ii}}$$

□

Multivariate logistic regression. We denote $y_j \in \{0,1\}^m$ the one-hot encoded label, $\hat{\Theta} \in \mathbb{R}^{p \times m}$ the learned parameter, and $\ell_j(\hat{\Theta}) = -\sum_{k=1}^{m} y_{jk} \log \hat{p}_{jk}$ the total loss, with $\hat{p}_j = \mathrm{softmax}(\hat{\Theta}^\top x_j)$ the predicted class probabilities. The results for the influence and Newton estimates are given in the following proposition.

Proposition B.3. *In the multivariate case for multinomial logistic regression over m classes, we have:*

$$\widehat{\Delta\ell}_i^{\mathrm{NS}} = g_i^\top H_{ii}(I_m - W_i H_{ii})^{-1} g_i \tag{21}$$

Where

- $g_i = \hat{p}_i - y_i \in \mathbb{R}^m$.
- $W_i = \mathrm{diag}(\hat{p}_i) - \hat{p}_i\hat{p}_i^\top \in \mathbb{R}^{m\times m}$.
- $H_{ii} = \tilde{X}_i^\top \mathcal{H}_{\hat{\Theta}}^{-1}\tilde{X}_i \in \mathbb{R}^{m\times m}$ *is the leverage matrix.*
- $\tilde{X}_i = x_i \otimes I_m \in \mathbb{R}^{pm\times m}$ *is the expanded feature matrix.*
- $\mathcal{H}_{\hat{\Theta}} = \sum_{j=1}^n x_j x_j^\top \otimes W_j \in \mathbb{R}^{pm\times pm}$ *is the full Hessian of the loss.*

Proof. Let $L(\hat{\Theta}) = \sum_{j=1}^n \ell_j(\hat{\Theta})$ be the total cross-entropy loss. For convenience, we parameterize the model by the vectorized weights $\hat{\Theta} \in \mathbb{R}^{pm}$ instead of a matrix $\in \mathbb{R}^{p\times m}$. Using the provided definitions, the per-sample gradient and Hessian evaluated at the optimal full-data parameters $\hat{\Theta}$ can be written compactly as:

$$\nabla_{\hat{\Theta}}\ell_i(\hat{\Theta}) = \tilde{X}_i g_i$$

$$\nabla^2_{\hat{\Theta}}\ell_i(\hat{\Theta}) = \tilde{X}_i W_i \tilde{X}_i^\top = (x_i \otimes I_m)W_i(x_i^\top \otimes I_m) = x_i x_i^\top \otimes W_i$$

Summing the per-sample Hessians gives the full Hessian $\mathcal{H}_{\hat{\Theta}} = \sum_{j=1}^n \tilde{X}_j W_j \tilde{X}_j^\top$. When we leave out the i-th sample, the leave-one-out Hessian $\mathcal{H}_{-i}$ is defined as:

$$\mathcal{H}_{-i} = \mathcal{H}_{\hat{\Theta}} - \tilde{X}_i W_i \tilde{X}_i^\top$$

Using a single Newton step approximation, the change in the estimated parameters upon removing the i-th sample is:

$$\Delta\hat{\Theta}_i \approx \mathcal{H}_{-i}^{-1}\nabla_\Theta \ell_i(\hat{\Theta}) = \mathcal{H}_{-i}^{-1}\tilde{X}_i g_i$$

The approximate change in the loss function for the left-out sample, $\Delta\ell_i$, evaluates the gradient against this parameter shift:

$$\Delta\ell_i \approx \nabla_\Theta \ell_i(\hat{\Theta})^\top \Delta\hat{\Theta}_i = (\tilde{X}_i g_i)^\top \mathcal{H}_{-i}^{-1}(\tilde{X}_i g_i) = g_i^\top \left(\tilde{X}_i^\top \mathcal{H}_{-i}^{-1}\tilde{X}_i\right) g_i$$

To evaluate the inner term without explicitly inverting the $pm \times pm$ LOO Hessian, we apply the Woodbury matrix identity to $\mathcal{H}_{-i}^{-1} = (\mathcal{H}_{\hat{\Theta}} - \tilde{X}_i W_i \tilde{X}_i^\top)^{-1}$. Because the weight matrix W_i is not strictly invertible (its columns sum to zero), we use the form of the Woodbury identity that does not require W_i^{-1}:

$$(\mathcal{H}_{\hat{\Theta}} - \tilde{X}_i W_i \tilde{X}_i^\top)^{-1} = \mathcal{H}_{\hat{\Theta}}^{-1} + \mathcal{H}_{\hat{\Theta}}^{-1}\tilde{X}_i W_i (I_m - \tilde{X}_i^\top \mathcal{H}_{\hat{\Theta}}^{-1}\tilde{X}_i W_i)^{-1}\tilde{X}_i^\top \mathcal{H}_{\hat{\Theta}}^{-1}$$

We substitute this expanded inverse into our target expression $\tilde{X}_i^\top \mathcal{H}_{-i}^{-1}\tilde{X}_i$. Leveraging the definition $H_{ii} = \tilde{X}_i^\top \mathcal{H}_{\hat{\Theta}}^{-1}\tilde{X}_i$, we obtain:

$$\tilde{X}_i^\top \mathcal{H}_{-i}^{-1}\tilde{X}_i = H_{ii} + H_{ii}W_i(I_m - H_{ii}W_i)^{-1}H_{ii}$$

Next, we factor out H_{ii} on the left side:

$$\tilde{X}_i^\top \mathcal{H}_{-i}^{-1}\tilde{X}_i = H_{ii}\left[I_m + W_i(I_m - H_{ii}W_i)^{-1}H_{ii}\right]$$

Applying the push-through matrix identity $W_i(I_m - H_{ii}W_i)^{-1} = (I_m - W_i H_{ii})^{-1}W_i$, the term inside the brackets becomes:

$$\tilde{X}_i^\top \mathcal{H}_{-i}^{-1}\tilde{X}_i = H_{ii}\left[I_m + (I_m - W_i H_{ii})^{-1}W_i H_{ii}\right]$$

To combine these terms, we multiply the identity matrix I_m by $(I_m - W_i H_{ii})^{-1}(I_m - W_i H_{ii})$:

$$\tilde{X}_i^\top \mathcal{H}_{-i}^{-1} \tilde{X}_i = H_{ii} \left[(I_m - W_i H_{ii})^{-1}(I_m - W_i H_{ii}) + (I_m - W_i H_{ii})^{-1} W_i H_{ii} \right]$$

Factoring out $(I_m - W_i H_{ii})^{-1}$ from both terms in the bracket leaves:

$$\tilde{X}_i^\top \mathcal{H}_{-i}^{-1} \tilde{X}_i = H_{ii}(I_m - W_i H_{ii})^{-1} \left(I_m - W_i H_{ii} + W_i H_{ii} \right)$$

The $-W_i H_{ii}$ and $+W_i H_{ii}$ cleanly cancel out, yielding the final simplified matrix expression:

$$\tilde{X}_i^\top \mathcal{H}_{-i}^{-1} \tilde{X}_i = H_{ii}(I_m - W_i H_{ii})^{-1}$$

Substituting this matrix back into our earlier equation for the scalar change in loss yields the desired formulation:

$$\widehat{\Delta \ell}_i^{\mathrm{NS}} = g_i^\top H_{ii}(I_m - W_i H_{ii})^{-1} g_i$$

□

Proposition B.4. *In the multivariate case for multinomial logistic regression over m classes, we have:*

$$\widehat{\Delta \ell}_i^{\mathrm{IF}} = g_i^\top H_{ii} g_i \tag{22}$$

Where g_i, H_{ii} are defined as in the previous proposition.

Proof. The influence function approximation is a special case of the Newton step approximation. This corresponds to replacing the matrix inverse $(I_m - W_i H_{ii})^{-1}$ with the identity matrix I_m. □

For $m = 2$ the per-sample weight collapses to the scalar w_j, $\mathcal{H}_{\hat{\Theta}} = X^\top W X$ factorizes as a single Kronecker product, and we recover the binary formulas of the previous subsection. In general $\mathcal{H}$ does *not* factorize as a Kronecker product because W_j depends on j, so the multinomial leverage cannot be separated into a sample term and a class term as in the binary case.

C Details on the Experimental Setup

C.1 Model Architectures and Training Hyperparameters

We evaluate our surrogates on a deliberately diverse benchmark spanning tabular regression, image classification, and text classification, with model sizes ranging from a single linear layer to a fully fine-tuned transformer. For each (dataset, architecture) pair, target and reference (shadow) models are trained with the *same* hyperparameters; only the random seed, which controls both weight initialization and the training split differs. We train 200 reference models and 16 target models per setting. Table 2 summarizes the architectures and Table 3 the training hyperparameters. The detailed implementation can be found in the codebase.

Table 2: Datasets and architectures used in our benchmark. $|\mathcal{D}|$ denotes the number of training examples used to draw membership splits. #classes is only relevant for classification tasks.

Dataset	Task	Model	Backbone	$\lvert\mathcal{D}\rvert$	#classes
UCI Energy	Reg.	MLP	2 hidden blocks [64, 32], dropout 0.3	614	—
California Housing	Reg.	Ridge Reg.	1 linear layer	16,512	—
California Housing	Reg.	MLP	3 hidden blocks [128, 128, 128]	16,512	—
BloodMNIST	Classif.	DINO	DINOv2 ViT-S/14 (frozen + linear head)	8,000	8
CIFAR-10	Classif.	CNN	3 conv blocks	50,000	10
CIFAR-100	Classif.	ResNet-20	3 stages, dropout 0.3	50,000	100
SST-2	Classif.	BERT	BERT-tiny (L=2, H=128)	67,349	2
SST-2	Classif.	BERT	BERT-base (frozen + linear head)	67,349	2

Table 3: Training hyperparameters. Loss is quadratic for regression and cross-entropy for classification.

Dataset	Model	Epochs	Batch	LR	Optimizer	Weight decay
UCI Energy	MLP	300	32	10^{-3}	Adam	10^{-3}
California Housing	Ridge Reg.	200	256	10^{-2}	Adam	10^{-3}
California Housing	MLP	200	256	10^{-3}	Adam	$5{\cdot}10^{-4}$
BloodMNIST	DINO (linear)	100	512	10^{-2}	SGD ($\mu=0.9$)	10^{-6}
CIFAR-10	CNN	25	256	10^{-3}	AdamW	$5{\cdot}10^{-4}$
CIFAR-100	ResNet-20	50	256	10^{-3}	SGD ($\mu=0.9$)	$5{\cdot}10^{-4}$
SST-2	BERT (full FT)	10	32	10^{-4}	AdamW	10^{-2}
SST-2	BERT (linear)	10	32	10^{-3}	AdamW	10^{-2}

Table 4: Compute budget for our experimental setting. Values are observed run-times in GPU-hours. *Surrogates* aggregates the GPU time spent computing the influence / Newton scores surrogates over all targets.

	Training			Attacks+Stats	Surrogates	IF (full network)	Total
Dataset / Model	per-model	200 refs	16 targets	200 refs	per-target	per-target	GPU-h
UCI Energy / MLP	30 s	1.7 h	8 min	30 s	3 s	–	1.7 h
California / Ridge reg.	1 min	3.3 h	0.3 h	7 min	10 s	–	3.6 h
California / MLP	1 min	3.3 h	0.3 h	7 min	10 s	20 min	8.9 h
BloodMNIST / DINO (linear)	15 min	50.0 h	4.0 h	5 min	15 s	–	54.0 h
CIFAR-10 / CNN	4 min	13.0 h	1.0 h	20 min	30 s	2.2 h	49.2 h
CIFAR-100 / ResNet-20	13 min	44.4 h	3.5 h	45 min	1 min	–	48.5 h
SST-2 / BERT-tiny (full FT)	6 min	20 h	1.6 h	17 min	45 s	–	20.6 h
SST-2 / BERT-base (linear)	19 min	64.6 h	5.0 h	18 min	3 min	–	70 h
ImageNet-100/ ResNet-50	43 min	143.3 h	11.2 h	36 min	6 min	–	191.8 h
Total							**447 GPU-h**

C.2 Time and Compute Resources

All experiments were run using V100 (32 GB) GPU. The values presented in Table 4 are the wall-clock times reported using dedicated packages. They split into four phases: (i) training the reference ensemble, (ii) training the target models, (iii) running the LiRA, RMIA and Shokri attacks plus the per-point statistics, and (iv) computing the leverage-score / influence / Newton surrogates on each target. We report the total time spent in each phase, as well as the total time for the entire benchmark.

To evaluate cost-effectiveness, surrogate performance is reported as the time required to compute scores for a single target, allowing a direct comparison with the training of the 200 reference models. While actual implementation involves computing scores across all targets, resulting in a total runtime approximately 16 times the per-target figure, the single-target metric highlights the efficiency gains. For example, using the DINO linear model, computing surrogates for one target takes only 15 seconds; this represents a $50{,}000\times$ speedup compared to the 60 hours required to train all the reference (shadow) models.

The training phase dominates the total cost. Once all models are trained, surrogates and attacks reuse the same models.

D Additional Experimental Results

Detailed results for the LiRA ASR top-$k\%$ recall are reported in Table 8. We observe that the Newton and Influence surrogates outperform the loss surrogate across most datasets and architectures. Although the Spearman correlation is often higher for our surrogates, the absolute values remain quite low (except for the CIFAR datasets).

This suggests that the ranking of samples by our surrogates is not perfectly aligned with the ranking by the LiRA score, mostly for the points that are in the middle of the distribution. This is not surprising as the LiRA ASR score is very noisy in this part of the distribution. This intuition is confirmed

by Figure 7: points in the middle of the distribution are scattered, whereas in the tails the ranking induced by the surrogates is much more consistent with the LiRA score.

D.1 Comparison with Full-Network Influence functions

To evaluate the impact of the last-layer approximation, we computed full-network influence functions for CIFAR-10 (CNN) and California Housing (MLP). These estimates were calculated using inverse Hessian-vector products [45] via the Conjugate Gradient (CG) algorithm [46]. We ensured CG convergence by setting a tolerance of 10^{-2} and applying a notably high damping value of 10; while strictly necessary for the algorithm to converge in our case, such heavy damping inherently alters the resulting influence estimates.

As presented in Table 4, the computational overhead for the full-network approach is substantial: for CIFAR-10, processing a single model took more than 2 hours, compared to only 30 seconds for the last-layer surrogate. Despite this significantly higher cost, the full-network influence function performed worse in identifying vulnerable samples. This suggests that the last-layer approximation not only offers a massive gain in efficiency but may also act as a useful regularizer by focusing on the most discriminative features. Furthermore, as noted by Basu et al. [16], this underperformance may come from the inherent fragility and poor scaling of influence functions when applied to the non-convex landscapes of deep networks.

Figure 3 illustrates the top-$k\%$ recall with the additional full network influence function surrogate $\nabla\ell_i^\top H_\theta^{-1}\nabla\ell_i$. The values are reported in the CIFAR-10 and California Housing (MLP) cells in Table 8.

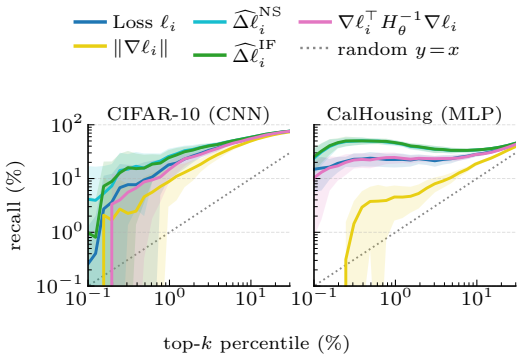

Figure 3: Recall of the $k\%$ most vulnerable to LiRA (ranked by the per-sample Attack Success Rate) when keeping the top-$k\%$ for each of the classical surrogates as well as the full Influence Function surrogate $\nabla\ell_i^\top H_\theta^{-1}\nabla\ell_i$. Axes are log-log; the dotted diagonal $y = x$ is the random baseline. Solid lines are means over 16 target models with ± 1 std bands.

D.2 Per-Sample TPR@low-FPR as Baseline

For completeness, we also evaluated the performance of all the proxies when using the per-sample TPR@low-FPR as the ground truth. We fixed each threshold to obtain a FPR of 1% and average the results over 16 target models.

Table 5 presents the Recall (%) of the ground-truth top 1% (ranked by per-sample TPR@FPR=1%) when selecting the top 5% according to each surrogate. Our surrogates still correlate better with this metric than the loss baseline.

D.3 Evaluation on Non-Members

Throughout our work, we framed all our surrogates from a member's point of view: how to estimate the loss gap $\ell_{-i} - \ell_i$, when the loss ℓ_i is available. We made this choice, as we believe it is the primary setting of interest. A model provider or auditor wants to assess which points in their datasets are vulnerable to costly MIAs.

One could instead ask to estimate the vulnerability of a point that is *not* in the training set, for instance before adding it. This corresponds to estimating the loss gap when only ℓ_{-i} is available. Since Proposition 3.4 gives a property of the data point itself, independent of its membership status, it can equally be used to derive an estimator of the loss gap for non-members, requiring a rank-one update rather than a downdate. For example, for linear regression, this yields:

$$\widehat{\Delta\ell_i}^{NS,+} = \frac{2e_{-i}^2 h_{-ii}}{1 + h_{-ii}}$$

with $h_{-ii} = x_i^\top G_{-i}^{-1} x_i$. There is an analogous correction for logistic regression.

To demonstrate this in practice, we computed our surrogates across several datasets for non-members. For each specific target model, all measures were computed exclusively on points held out from its training set. The results in Table 6 confirm that our estimators also work in identifying the most vulnerable non-members.

D.4 Evaluation on a Larger Dataset

We additionally evaluated a ResNet-50 [34] classification model trained on ImageNet-100 [47] (restriction of ImageNet to 100 classes). Due to the large size of the last layer, we used a K-FAC approximation [44] to compute the scores. We trained 200 shadow models and 16 target models using a standard 45-epoch schedule. At this rate, depending on the seed, some models interpolated, yielding up to 99% training accuracy. We present in Table 7 the results averaged over 16 target models, with the LiRA ASR as target metric.

In this specific case, our estimates performed worse than the loss baseline. We believe this discrepancy stems from three possible factors:

- The interpolation regime, which requires a different analysis since many samples are highly vulnerable. ImageNet-100 is by far the most vulnerable dataset in our suite (mean per-sample LiRA ASR of 0.83, compared to 0.77, 0.68, and 0.57 for CIFAR-100, CIFAR-10, and BloodMNIST, respectively). As a model approaches interpolation, the residual training loss ranks the margin rather than the learning difficulty, and all loss-derived proxies degrade.
- The leverage score carries less information in this interpolating regime. It has a Spearman rank correlation with the loss of 0.91 (compared to 0.59 for CIFAR-100), meaning they contain the same information in terms of vulnerability.
- The use of the K-FAC approximation can also alter the results.

D.5 Top-k Recall and Spearman Correlation for LiRA Attack

We report full results for the LiRA MIA in Table 8. Notably, we report the Spearman rank correlation between the surrogate scores and the per-point ASR for reference, but we do not find a strong correlation. We attribute this to the fact that points near the center of the vulnerability distribution yield an Attack Success Rate (ASR) close to 0.5, offering little informative value. In contrast, top-k recall effectively isolates the highly vulnerable points, which are the primary focus for practical security auditing and defense.

For classification models we also report the self-entropy defined as $\mathrm{H}_i = -\sum_{k=1}^{m} \hat{p}_{i,k} \log \hat{p}_{i,k}$, where $\hat{p}_{i,k}$ is the model's predicted probability of class k on input x_i. Self-entropy is a standard confidence-based MIA baseline [48, 18]: training samples typically receive more peaked predictions and hence lower entropy than non-members.

D.6 Top-k Recall and Spearman Correlation for RMIA Attack and Classifier Shokri's Attack

The RMIA [10] attack is computed in the online setting, while the classifier-based [3] approach uses a Random Forest [49] classifier trained with the logits of the target model as input. We provide the top-k recall curves for RMIA and Shokri's attack in Figure 4 and Figure 5 respectively.

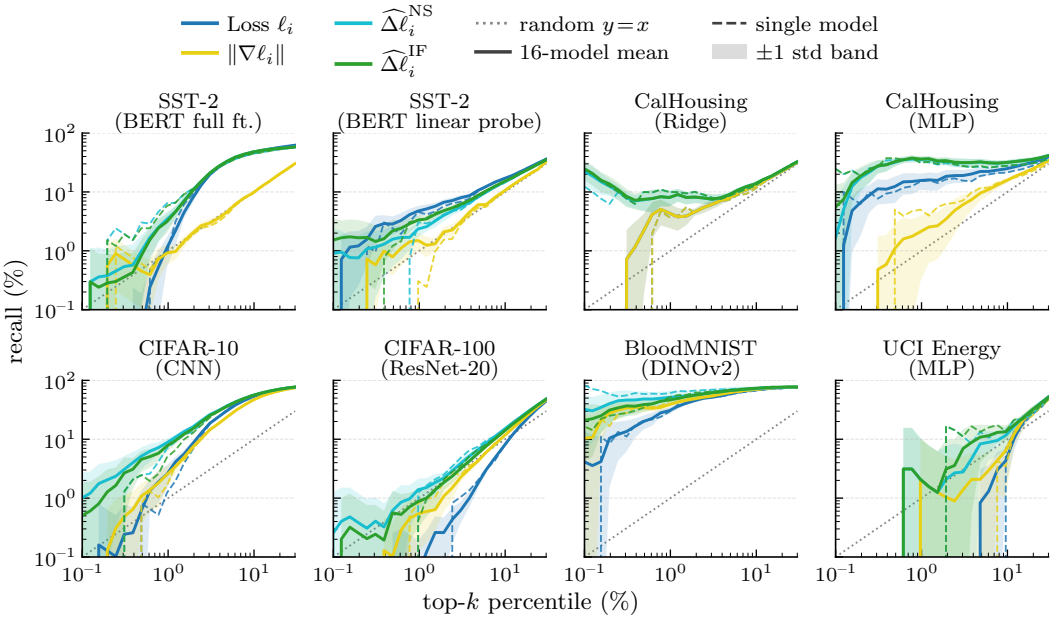

Figure 4: Recall of the $k\%$ most vulnerable to **RMIA** (ranked by the per-sample Attack Success Rate) when keeping the top-$k\%$ for each of the four surrogates: the loss ℓ_i, the gradient norm $\|\nabla \ell_i\|$, the influence estimate $\widehat{\Delta\ell}_i^{\mathrm{IF}}$, and the Newton estimate $\widehat{\Delta\ell}_i^{\mathrm{NS}}$. Axes are log-log; the dotted diagonal $y = x$ is the random baseline. Solid lines are means over 16 target models with ± 1 std bands; dashed-lines show a single representative model.

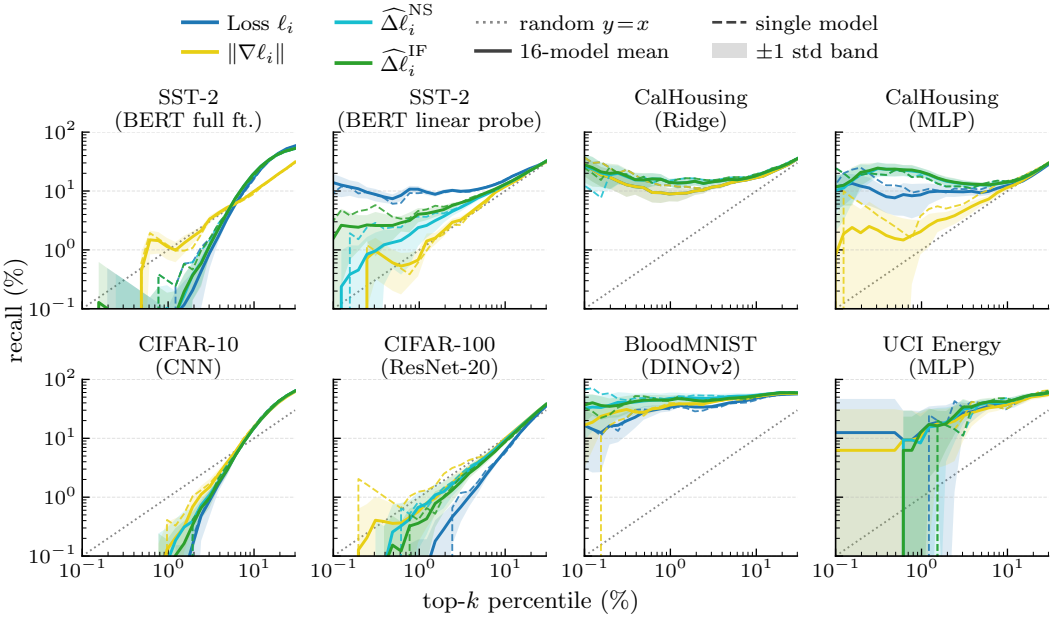

Figure 5: Recall of the $k\%$ most vulnerable to Shokri's attack (ranked by the per-sample Attack Success Rate) when keeping the top-$k\%$ for each of the four surrogates: the loss ℓ_i, the gradient norm $\|\nabla \ell_i\|$, the influence estimate $\widehat{\Delta\ell}_i^{\mathrm{IF}}$, and the Newton estimate $\widehat{\Delta\ell}_i^{\mathrm{NS}}$. Axes are log-log; the dotted diagonal $y = x$ is the random baseline. Solid lines are means over 16 target models with ± 1 std bands; dashed-lines show a single representative model.

D.7 Additional Visualizations

We provide additional visualizations of the samples with the highest and lowest influence scores and Newton scores for the BloodMNIST, CIFAR-10 and CIFAR-100 datasets in Figure 6. We also provide the top-10 highest and lowest influence scores and Newton scores for the BERT model trained on the SST-2 dataset with linear probing in Table 9. The joint distribution of local loss ℓ_i and influence score for every training sample of a single target model, colored by per-sample LiRA ASR, is presented in Figure 7.

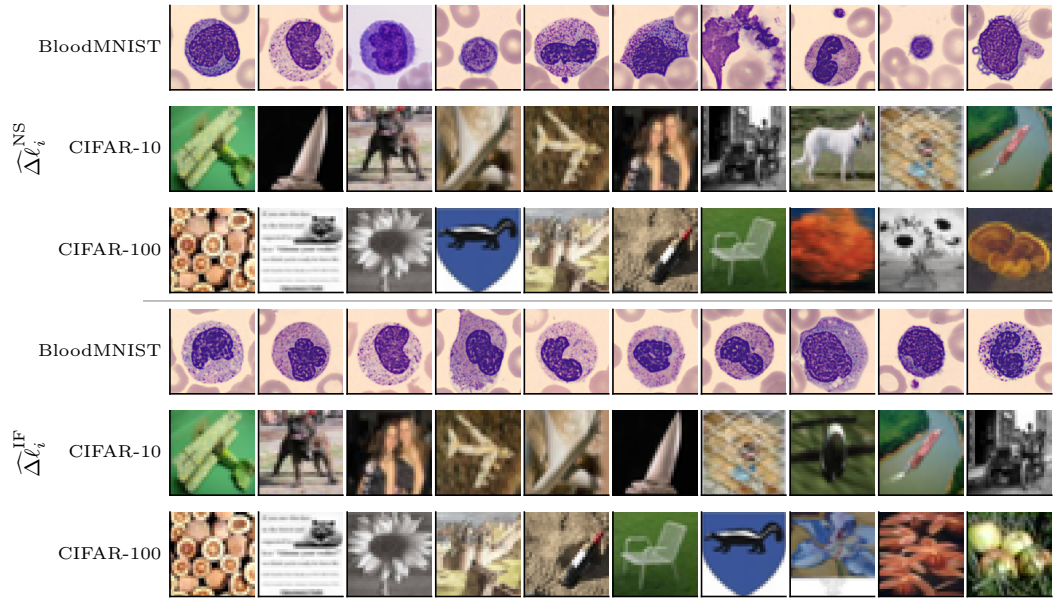

Figure 6: Visualization of the 10 images with the highest influence and Newton scores for BloodMNIST (top), CIFAR-10 (middle) and CIFAR-100 (bottom).

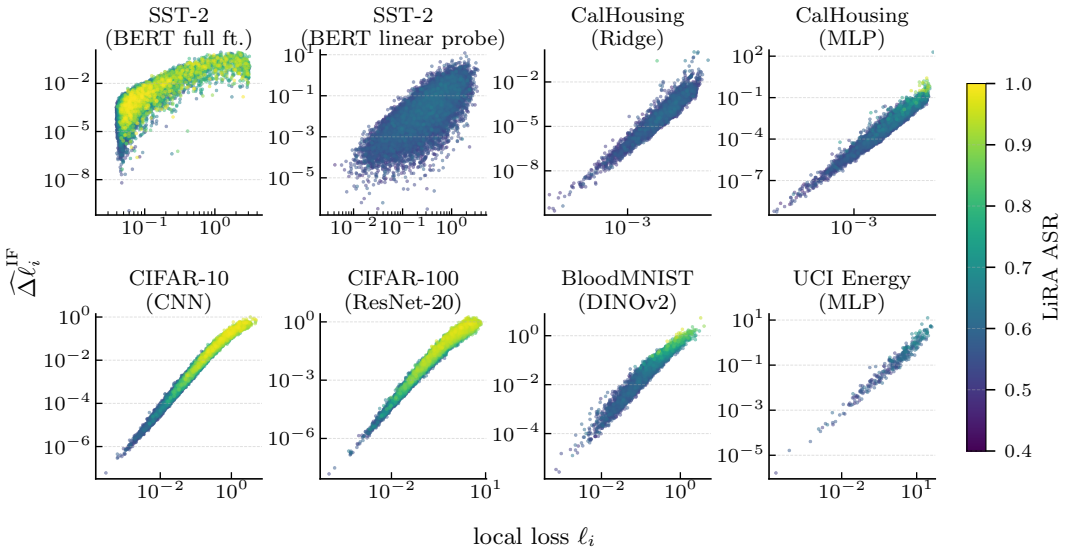

Figure 7: Joint distribution of local loss ℓ_i and influence for every training sample of a single target model, coloured by per-sample LiRA ASR, log-log axes.

Table 5: Recall of the ground-truth top 1% most vulnerable samples (ranked by LiRA **per-sample TPR@FPR=1%**) when selecting the top 5% according to each surrogate. Mean ± std across target models. Higher is better; **bold** = best per dataset.

		Recall of Top 1% @5% (in %)		
Dataset	Model	Loss ℓ_i	$\widehat{\Delta\ell_i}^{\mathrm{NS}}$	$\widehat{\Delta\ell_i}^{\mathrm{IF}}$
SST-2	BERT, full ft.	37.1 ± 2.6 %	37.1 ± 2.9 %	**37.2 ± 2.9 %**
CIFAR-10	CNN	58.2 ± 1.7 %	**67.0 ± 1.7 %**	66.4 ± 1.7 %
CIFAR-100	ResNet-20	26.7 ± 2.5 %	**37.5 ± 5.3 %**	37.1 ± 5.1 %
BloodMNIST	DINOv2	78.6 ± 4.8 %	86.0 ± 11.2 %	**86.5 ± 6.6 %**

Table 6: Recall of the ground-truth top 1% most vulnerable samples (ranked by LiRA per-sample ASR) when selecting the top 5% according to each surrogate for points that were **out** of the training set. Mean ± std across target models. Higher is better; **bold** = best per dataset.

		Recall of Top 1% @5% (in %)			
Dataset	Model	Loss ℓ_{-i}	$\|\nabla\ell_{-i}\|$	Newton out	Influence out
CalHousing	Ridge regression	**12.1 ± 2.5**	**12.1 ± 2.5**	11.3 ± 2.6	11.3 ± 2.6
CalHousing	MLP	57.1 ± 9.4	22.7 ± 7.3	79.8 ± 5.5	**80.3 ± 5.2**
CIFAR-10	CNN	74.6 ± 2.5	30.8 ± 2.1	**78.8 ± 2.6**	78.7 ± 2.5
CIFAR-100	ResNet-20	43.2 ± 2.5	10.1 ± 1.9	**47.8 ± 6.2**	46.9 ± 6.1
BloodMNIST	DINOv2	85.7 ± 4.3	88.1 ± 3.1	95.9 ± 2.5	**98.1 ± 1.2**
UCI Energy	tab. MLP	25.0 ± 33.3	18.8 ± 32.1	**54.2 ± 31.9**	**54.2 ± 31.9**

Table 7: Recall of the ground-truth top 1% most vulnerable samples (ranked by LiRA per-sample ASR) when selecting the top 5% according to each surrogate for the **ImageNet** dataset. Mean ± std across target models. Higher is better; **bold** = best per dataset.

Dataset	Model	Loss ℓ_i	$\|\nabla\ell_i\|$	$\widehat{\Delta\ell_i}^{\mathrm{NS}}$	$\widehat{\Delta\ell_i}^{\mathrm{IF}}$
ImageNet-100	ResNet-50	**22.14 ± 5.38 %**	18.78 ± 4.35 %	18.31 ± 3.80 %	18.63 ± 3.79 %

Table 8: LiRA membership-inference attack: top-k recall (%) and Spearman rank correlation. Best results per dataset are in **bold**. In contrast to Table 1, the recall values here are computed by keeping the top-k samples with the highest surrogate scores and checking how many of them are in the top-k samples with the highest LiRA ASR score.

Dataset	Surrogate	Recall (%) @0.1%	Recall (%) @1%	Recall (%) @5%	Spearman ρ
SST-2 (BERT, full ft.)	Loss ℓ_i	0.00 ± 0.00	5.44 ± 1.36	30.83 ± 1.11	$\mathbf{0.474 \pm 0.022}$
	$\|\nabla \ell_i\|$	0.00 ± 0.00	0.57 ± 0.18	4.64 ± 0.33	–
	Self-entropy	0.02 ± 0.06	6.34 ± 1.16	28.52 ± 1.10	0.470 ± 0.022
	$\widehat{\Delta\ell_i}^{\mathrm{NS}}$	1.02 ± 1.47	$\mathbf{7.05 \pm 1.23}$	$\mathbf{31.30 \pm 1.44}$	0.389 ± 0.036
	$\widehat{\Delta\ell_i}^{\mathrm{IF}}$	$\mathbf{1.22 \pm 1.54}$	6.88 ± 1.30	31.28 ± 1.44	0.389 ± 0.036
SST-2 (BERT, linear probe)	Loss ℓ_i	0.05 ± 0.08	$\mathbf{2.30 \pm 0.64}$	$\mathbf{7.94 \pm 0.50}$	$\mathbf{0.045 \pm 0.005}$
	$\|\nabla \ell_i\|$	0.00 ± 0.00	1.56 ± 0.70	6.64 ± 0.65	–
	Self-entropy	$\mathbf{0.60 \pm 1.26}$	1.07 ± 0.46	5.60 ± 0.34	0.026 ± 0.005
	$\widehat{\Delta\ell_i}^{\mathrm{NS}}$	0.00 ± 0.00	1.80 ± 0.87	6.88 ± 0.57	0.037 ± 0.005
	$\widehat{\Delta\ell_i}^{\mathrm{IF}}$	0.30 ± 0.94	2.27 ± 0.70	7.01 ± 0.43	0.038 ± 0.005
CalHousing (ridge)	Loss ℓ_i	6.33 ± 7.74	3.16 ± 1.81	8.65 ± 1.05	0.033 ± 0.006
	$\|\nabla \ell_i\|$	6.33 ± 7.74	3.16 ± 1.81	8.65 ± 1.05	0.033 ± 0.006
	$\widehat{\Delta\ell_i}^{\mathrm{NS}}$	36.44 ± 15.34	$\mathbf{7.54 \pm 2.50}$	9.30 ± 1.23	0.050 ± 0.007
	$\widehat{\Delta\ell_i}^{\mathrm{IF}}$	$\mathbf{38.02 \pm 14.64}$	$\mathbf{7.54 \pm 2.50}$	$\mathbf{9.30 \pm 1.24}$	$\mathbf{0.050 \pm 0.007}$
CalHousing (MLP)	Loss ℓ_i	14.90 ± 9.25	22.59 ± 6.49	23.58 ± 4.03	0.276 ± 0.028
	$\|\nabla \ell_i\|$	0.00 ± 0.00	4.53 ± 2.83	10.81 ± 2.26	0.198 ± 0.019
	$\widehat{\Delta\ell_i}^{\mathrm{NS}}$	23.95 ± 11.89	45.84 ± 5.54	33.22 ± 3.96	0.297 ± 0.027
	$\widehat{\Delta\ell_i}^{\mathrm{IF}}$	$\mathbf{25.49 \pm 11.97}$	$\mathbf{46.46 \pm 4.79}$	$\mathbf{33.46 \pm 3.88}$	$\mathbf{0.297 \pm 0.027}$
	Full influence	10.35 ± 9.35	24.57 ± 5.55	23.72 ± 2.85	0.266 ± 0.025
CIFAR-10 (CNN)	Loss ℓ_i	4.02 ± 3.23	17.30 ± 2.10	42.98 ± 1.17	0.813 ± 0.004
	$\|\nabla \ell_i\|$	0.01 ± 0.05	8.40 ± 1.65	33.52 ± 1.28	0.811 ± 0.003
	Self-entropy	0.78 ± 1.57	10.36 ± 1.33	38.70 ± 1.35	0.808 ± 0.004
	$\widehat{\Delta\ell_i}^{\mathrm{NS}}$	$\mathbf{15.28 \pm 6.95}$	$\mathbf{25.59 \pm 1.77}$	$\mathbf{48.14 \pm 1.27}$	0.814 ± 0.003
	$\widehat{\Delta\ell_i}^{\mathrm{IF}}$	14.07 ± 7.06	24.48 ± 1.99	47.76 ± 1.31	0.814 ± 0.003
	Full influence	0.00 ± 0.00	14.94 ± 6.07	41.49 ± 3.19	$\mathbf{0.821 \pm 0.009}$
CIFAR-100 (ResNet-20)	Loss ℓ_i	2.26 ± 2.86	7.40 ± 1.80	20.53 ± 0.72	0.658 ± 0.007
	$\|\nabla \ell_i\|$	0.51 ± 1.39	2.68 ± 0.69	13.94 ± 0.93	0.659 ± 0.005
	Self-entropy	0.25 ± 0.99	3.84 ± 1.15	19.75 ± 1.11	0.686 ± 0.007
	$\widehat{\Delta\ell_i}^{\mathrm{NS}}$	4.84 ± 4.60	$\mathbf{13.51 \pm 2.82}$	$\mathbf{28.64 \pm 3.15}$	$\mathbf{0.690 \pm 0.024}$
	$\widehat{\Delta\ell_i}^{\mathrm{IF}}$	$\mathbf{5.77 \pm 3.82}$	13.08 ± 2.66	28.24 ± 2.80	0.689 ± 0.022
BloodMNIST (DINOv2)	Loss ℓ_i	10.55 ± 10.08	33.22 ± 3.72	59.06 ± 2.19	0.600 ± 0.005
	$\|\nabla \ell_i\|$	10.91 ± 14.54	39.95 ± 4.54	62.34 ± 2.07	0.604 ± 0.005
	Self-entropy	0.00 ± 0.00	14.67 ± 3.72	44.45 ± 2.05	0.586 ± 0.005
	$\widehat{\Delta\ell_i}^{\mathrm{NS}}$	$\mathbf{34.70 \pm 26.66}$	$\mathbf{52.68 \pm 15.24}$	67.00 ± 10.01	0.604 ± 0.031
	$\widehat{\Delta\ell_i}^{\mathrm{IF}}$	27.24 ± 17.77	47.27 ± 7.87	$\mathbf{67.09 \pm 6.55}$	$\mathbf{0.608 \pm 0.024}$
UCI Energy (tab. MLP)	Loss ℓ_i	0.00 ± 0.00	2.01 ± 8.06	20.74 ± 4.87	0.528 ± 0.040
	$\|\nabla \ell_i\|$	0.00 ± 0.00	6.25 ± 12.92	26.83 ± 9.21	0.507 ± 0.036
	$\widehat{\Delta\ell_i}^{\mathrm{NS}}$	0.00 ± 0.00	10.90 ± 19.74	28.25 ± 7.50	$\mathbf{0.541 \pm 0.032}$
	$\widehat{\Delta\ell_i}^{\mathrm{IF}}$	$\mathbf{6.25 \pm 25.00}$	$\mathbf{14.72 \pm 23.99}$	$\mathbf{30.32 \pm 7.89}$	0.540 ± 0.032

Table 9: Top-10 highest-|leverage| training sentences for BERT (linear probe) on SST-2, ranked by Newton (left) and Influence (right) surrogates.

Rank	$\widehat{\Delta\ell_i}^{\mathrm{NS}}$	$\widehat{\Delta\ell_i}^{\mathrm{IF}}$
1	to appearing in this junk that 's tv sitcom material at best	is never quite able to overcome the cultural moat surrounding its l...
2	the filmmakers wisely decided to let crocodile hunter steve irwin d...	spider-man
3	the kind that pretends to be passionate and truthful but is really ...	nicks refuses to let slackers be seen as just another teen movie , ...
4	a supernatural mystery that does n't know whether it wants to be a ...	recycle images and characters that were already tired 10 years ago
5	less the cheap thriller you 'd expect than it is a fairly revealing...	report that the children of varying ages in my audience never cough...
6	played-out	breach gaps in their relationships with their fathers
7	of a callow rich boy who is forced to choose between his beautiful ...	it 's painful to watch witherspoon 's talents wasting away inside u...
8	that collect a bunch of people who are enthusiastic about something...	its drastic iconography
9	: the widowmaker is derivative , overlong , and bombastic – yet su...	of a callow rich boy who is forced to choose between his beautiful ...
10	ride around a pretty tattered old carousel .	never does " lilo & stitch " reach the emotion or timelessness of...

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The three contributions stated in the introduction, (i) the closed-form decomposition of the expected leave-one-out gap into the leverage $\bar{h}_{ii}$ and residual ε_i, (ii) the influence-function and Newton-step surrogates with closed-form last-layer expressions, and (iii) the top-k recall results on $7/8$ dataset/architecture pairs against LiRA, RMIA and the shadow-model classifier attack, are established respectively in Sections 3, 4 and 5.1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the work are discussed in the Conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All theoretical statements (Lemma 3.1, Proposition 3.4, Corollary 3.3) explicitly state the conditioning on the target point and the Gaussian linear model assumptions. Complete proofs are given in Appendix A; the derivations for the influence-function and Newton-step estimators in the generalized-linear case are given in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 5.1 lists the eight datasets/architectures, the LiRA member/non-member split protocol, the number of shadow ($K = 200$) and target (16) models, the optimizers used per architecture, and the surrogates being compared. Full hyper-parameters are given in Appendix C.1 and the supplementary code archive contains the launch scripts used to reproduce all reported numbers.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All datasets used (CIFAR-10/100, BloodMNIST, SST-2, California Housing, UCI Energy) are publicly available in a github repository contains training, attack, scoring and figure scripts.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5.1 reports the data splits (LiRA member/non-member partitioning with per-model seeds), the optimizers used per architecture (AdamW, SGD-momentum, Adam) and the standard recipes followed. The complete list of learning rates, batch sizes, weight decay and number of epochs is provided in Appendix C.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Table 1 reports recall as mean $\pm$ standard deviation across the 16 independent target models, and Figure 2 shows ± 1 standard-deviation bands around the mean of the top-k recall, with the source of variability (random member/non-member splits and target-model initialization) made explicit in Section 5.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 5.1 reports that all experiments were run on a V100 GPU cluster, and indicates the compute scale set by training $K = 200$ shadow models plus 16 target models per dataset/architecture. Per-job resource and runtime details are listed in the appendix C.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses only public benchmark datasets and does not involve human subjects, personal data collection, or deployed systems; it conforms with the NeurIPS Code of Ethics. Anonymity was preserved in the submission.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper is motivated by the positive impact of enabling cheaper individual privacy-risk auditing, complementing differential-privacy guarantees with a per-sample diagnostic. This technique could not a priori be used directly as an attack, as it relies on accessing the full training dataset to compute the scores.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release new pretrained generative models, scraped datasets, or other high-misuse-risk artifacts. The released code consists of training, attack and scoring scripts on public benchmark datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets (CIFAR-10/100, BloodMNIST, SST-2, California Housing, UCI Energy) and pretrained backbones (BERT, DINOv2 ViT-small) are publicly available and cited at first use; the attacks we compare against (LiRA, RMIA, the original shadow-model attack) are also cited. We use these assets in accordance with their respective licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The only new asset is the code implementing the computation of Influence and Newton-step scores for the last-layer, both for quadratic and cross-entropy loss. The code and the whole experimental pipeline are included. The supplementary archive includes a README documenting how to reproduce all reported results.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects, so no IRB approval was required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the core methodology. The methods, theoretical results and experiments do not involve LLMs as an important, original, or non-standard component; any LLM use was limited to writing or editing assistance.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.