

Leave-One-Out Distinguishability in Machine Learning

Jiayuan Ye[†], Anastasia Borovykh[‡], Soufiane Hayou[†], and Reza Shokri^{†*}

[†] National University of Singapore, [‡] Imperial College London

October 2, 2023

Abstract

We introduce a new analytical framework to quantify the changes in a machine learning algorithm’s output distribution following the inclusion of a few data points in its training set, a notion we define as leave-one-out distinguishability (LOOD). This problem is key to measuring data **memorization** and information **leakage** in machine learning, and the **influence** of training data points on model predictions. We illustrate how our method broadens and refines existing empirical measures of memorization and privacy risks associated with training data. We use Gaussian processes to model the randomness of machine learning algorithms, and validate LOOD with extensive empirical analysis of information leakage using membership inference attacks. Our theoretical framework enables us to investigate the causes of information leakage and where the leakage is high. For example, we analyze the influence of activation functions, on data memorization. Additionally, our method allows us to optimize queries that disclose the most significant information about the training data in the leave-one-out setting. We illustrate how optimal queries can be used for accurate **reconstruction** of training data.

1 Introduction

A key question in interpreting a model involves identifying which members of the training set have *influenced* the model’s predictions for a particular query [15, 16, 26]. Our goal then is to understand the impact of the addition of a specific data point into the training set on the model’s predictive behavior. This becomes even more important when predictions on a data point are primarily due to its own presence (or that of points similar to it) in the training set - a phenomenon referred to as *memorization* [6]. Memorization in turn can cause leakage of sensitive information about the training data. This enables an adversary to infer which data points were included in the training set (using *membership inference attacks*), by observing the model’s predictions [30]. One way to address such questions is by training models with various combinations of data points, and subsequently measuring their **influence**, **memorization**, and **leakage**. This empirical approach, however, is computationally intense [7] and does not enable further efficient analysis of, for example, what data points are influential, which queries they influence the most, and what properties of model (architectures) can impact their influence. These limitations demand an efficient and constructive modeling approach to influence and leakage analysis. This is the focus of the current paper.

We unify the above-mentioned closely-related concepts, and define them as the statistical divergence between the output distribution of models, trained using (stochastic) machine learning algorithms, when

* Authors AB, SH and RS are ordered alphabetically.

one or more data points are added to (or removed from) the training set. We refer to this measure as leave-one-out distinguishability (LOOD). A greater LOOD on a data point S in relation to query Q suggests a larger influence of S on Q and a larger potential for information leakage about S when an adversary observes the prediction of models trained on S and queried at Q . We aim to examine which query points are most influenced by any S to provide a quantitative assessment of the leakage when given black box access to models. Furthermore, to gauge the severity of this leakage, we want to determine to what extent this leaked information can be harnessed to reconstruct the original training data. We use LOOD to provide both analytical and algorithmic solutions to these questions. The analytical structure of LOOD should also guide us in understanding how the model architecture, such as the choice of activation functions, can affect information leakage. These are the core research questions we seek to address.

We construct our analytical model using Gaussian Processes (GPs) to model the output distribution of neural networks in response to queries on any given set of data points. We also use Neural Network Gaussian Processes (NNGPs) to reason about LOOD. NNGPs are widely used in the literature as standalone models [23], as modeling tools for different types of neural networks [18, 24, 13, 33], or as building blocks for model architecture search [25] or data distillation for generating high-utility data [19]. We empirically validate that computed LOOD under NNGP models correlates well with the performance of membership inference attacks [35, 3] against neural networks in the leave-one-out setting for benchmark datasets (Figure 3a). This demonstrates that our approach allows for an accurate assessment of *information leakage* for deep neural networks, while significantly reducing the required computation time (by more than two orders of magnitude - Footnote 4) as LOOD under NNGP can be analytically computed. As a special form of LOOD, if we measure the gap between the *mean* predictions of the leave-one-out models on a query, instead of quantifying the divergence between their *distributions*, we can estimate the influence [15, 16] and memorization (self-influence) [7, 6] metrics (Section 4.3). We experimentally show that mean distance LOOD under NNGP agrees with the actual prediction differences under leave-one-out NN retraining (Figure 3c), thus faithfully capturing self-influence or memorization for deep neural networks.

Given our framework, we prove that, for various kernel functions, including the RBF kernel and NNGP kernel on a sphere (Theorem 3.1), the differing data point itself is a stationary point of the leave-one-out distinguishability. Through experiments we further show that on image datasets, LOOD under NNGP model is maximized when the query equals the differing datapoint (2b). We then experimentally show that this conclusion generalizes to deep neural networks trained via SGD, by showing that the empirical information leakage is maximized when query data is within a small perturbation of the differing data (Figure 2c). What is interesting is that, we empirically show that LOOD remains almost unchanged even as the number of queries increases (Appendix E.1). In other words, all the information related to the differing point in the leave-one-out setting is contained within the model’s prediction distribution solely on that differing point. We show how to exploit this to reconstruct training (image) data in the leave-one-out or leave-one group-out settings. We present samples of the reconstructed images (See Figure 10, and 15 through 19) which show significant resemblance to the training data, even when our optimization problem lands on sub-optimal queries.

Finally, we investigate how LOOD is affected by activation functions. We prove that a low-rank kernel matrix implies low LOOD, thus low information leakage (See Proposition 5.1). Leveraging existing literature, we further show that the NNGP kernel for fully connected networks is closer to a low-rank matrix under ReLU activation (thus lead to less leakage) compared to smooth activation functions such as GeLU. We also validate this effect of activation choice result empirically, both for NNGP and for deep neural network trained via SGD (Figure 5).

2 Definition and the Analytical Framework

Definition 2.1 (Leave-one-out distinguishability). Let D and D' be two training datasets that differ in record(s) S . Let $f|_D(Q)$ be the distribution of model predictions given training dataset D and query dataset Q , where the probability is taken over the randomness of the training algorithm. We define leave-one-out distinguishability as the statistical distance between $f|_D(Q)$ and $f|_{D'}(Q)$:

$$LOOD(Q; D, D') := \text{dist}(f|_D(Q), f|_{D'}(Q)). \quad (1)$$

In this paper, we mainly use the KL divergence [17] as the distance function dist to capture the mean and covariance matrix of prediction distribution. We also use the distance between the mean predictions of $f|_D(Q)$ and $f|_{D'}(Q)$ which we refer to as the Mean Distance LOOD. Different distance functions allow us to recover information leakage, memorization, and influence; see Section 4 for more details. We refer to Appendix C for a detailed discussion of why KL divergence is better suited than mean distance for measuring information leakage.

Gaussian process. To analytically capture the randomness of model outputs, without the need for empirical computation through multiple re-training, we use Gaussian Processes (GPs). A GP $f \sim GP(\mu, K)$ is specified by its mean prediction function $\mu(\mathbf{x}) = \mathbb{E}_f[f(\mathbf{x})]$ for any input \mathbf{x} , and covariance kernel function $K(\mathbf{x}, \mathbf{x}') = \mathbb{E}_f[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))]$ for any pair of inputs \mathbf{x} and \mathbf{x}' [32]. The mean function μ is typically set to e.g. zero, linear or orthogonal polynomial functions, and the covariance kernel function K completely captures the structure of a Gaussian process. Some commonly used kernel functions are RBF kernels $g(z) = \exp(-\frac{z}{2l})$ with length l , and correlation kernels $g\left(\frac{\langle \mathbf{x}, \mathbf{x}' \rangle}{\|\mathbf{x}\| \|\mathbf{x}'\|}\right)$. The posterior prediction distribution for GP given training data D (with labels y_D) when computed on a query data point Q follows the multivariate Gaussian $f|_{D, \sigma^2}(Q) \sim \mathcal{N}(\mu_{D, \sigma^2}(Q), \Sigma_{D, \sigma^2}(Q))$ with analytical expressions for its mean $\mu_{D, \sigma^2}(Q) = K_{QD}(K_{DD} + \sigma^2 \mathbb{I})^{-1} y_D$ and its covariance matrix $\Sigma_{D, \sigma^2}(Q) = K_{QQ} - K_{QD}(K_{DD} + \sigma^2 \mathbb{I})^{-1} K_{DQ}$.¹ Here for brevity, we denote by K_{XZ} the $p \times k$ kernel matrix given by $K_{XZ} = (K(\mathbf{x}_i, \mathbf{z}_j))_{1 \leq i \leq p, 1 \leq j \leq k}$, for $X = (\mathbf{x}_1, \dots, \mathbf{x}_p)$ and $Z = (\mathbf{z}_1, \dots, \mathbf{z}_k)$. Additionally, σ^2 represents the noise in the labels: $y_D = f(D) + \mathcal{N}(0, \sigma^2 \mathbb{I})$.

Neural network Gaussian process and its connections to NN training. Prior work has shown that the output of neural networks on a query converges to a Gaussian process, at initialization as the width of the NN tends to infinity [18, 29, 28]. The kernel function of the neural network Gaussian process (NNGP) is given by $K(\mathbf{x}, \mathbf{x}') = \mathbb{E}_{\theta \sim \text{prior}} [\langle f_\theta(\mathbf{x}), f_\theta(\mathbf{x}') \rangle]$, where θ refers to the neural network weights (sampled from the *prior* distribution), and f_θ refers to the prediction function of the model with weights θ . In this case, instead of the usual gradient-based training, we can analytically compute the Gaussian Process regression posterior distribution of the network prediction given the training data and query inputs. See Appendix B for more discussions on the connections between NNGP and the training of its corresponding neural networks.

We analytically compute and analyze LOOD on NN models using the NNGP framework. We also empirically validate our theoretical results to show the NNGP-computed insights generalise to NNs.

3 Optimizing LOOD to Identify the Most Influenced Point

LOOD (1) measures the influence of training data $S \in D$ on the prediction at record Q , with a higher magnitude reflecting more influence. In this section we analyse *where (at which Q) the model is most*

¹This is known as Gaussian Process regression.

influenced by the change of one train datapoint S . Besides shedding light on the fundamental question of influence [15] (and from there connections to adversarial robustness [8]), this question is also closely related to the open problem of constructing a worst-case query for tight privacy auditing via membership inference attacks (MIAs) [14, 21, 31, 22].

In this section, we consider a *single* query ($m = 1$) and single differing point ($|S| = 1$) setting and analyse the most influenced point by maximizing the LOOD:

$$\arg \max_Q LOOD(Q; D, D') := dist(f|_{D, \sigma^2}(Q), f|_{D', \sigma^2}(Q)), \quad (2)$$

both theoretically (via characterizing the landscape of the LOOD objective) and empirically (via solving problem (2) through gradient descent). Our analysis also extends generally to the multi-query ($|Q| > 1$) and a group of differing data ($|S| > 1$) setting – see Appendix E for the details.

We first prove that for a wide class of kernel functions, the differing point is a stationary point for the LOOD maximization objective (2), i.e., satisfies the first-order optimality condition. We then validate through experimental results on image data that the differing point is the global maximum and discuss one setting where this is no longer the case.

Analysis: Differing point is a stationary point for LOOD. We first prove that if the kernel function satisfies certain regularity conditions, the gradient of LOOD at the differint point is zero, i.e., the differing data is a stationary point for LOOD (as a function of query data).

Theorem 3.1. *Let the kernel function be K . Assume that (i) $K(x, x) = 1$ for all training data x , i.e., all training data are normalized under kernel function K ; (ii) $\frac{\partial K(x, x')}{\partial x} |_{x'=x} = 0$ for all training data x , i.e., the kernel distance between two data records x and x' is stable when they are equal. Then, the differing data is a stationary point for LOOD as a function of query data, as follows*

$$\nabla_Q LOOD(f|_{D, \sigma^2}(Q) \| f|_{D', \sigma^2}(Q)) |_{Q=S} = 0.$$

The full proof of Theorem 3.1 is deferred in Appendix D.1. Note that Theorem 3.1 requires the kernel function K to satisfy some regularity conditions. In Appendix D.2 we show the commonly used RBF kernel and NNGP kernel on a sphere satisfy these regularity conditions.

Theorem 3.1 proves that the first-order stationary condition of LOOD is satisfied when the query equals differing data. This is a necessary but not sufficient condition for maximal information leakage when query equals the differing data. However, as LOOD is a non-convex objective, it is hard to establish global optimality theoretically. Therefore, below we conduct extensive numerical experiments for Gaussian processes and deep neural networks to complement our theoretical results.

Experimental results for maximal information leakage when query equals differing data. We empirically run gradient descent on the LOOD objective (2) to find the single query that is most affected by the differing point. Figure 1 shows the optimized query under the RBF kernel. More examples of the optimized query under NNGP kernel functions are in Appendix D.3.

Next we present more evidence for the empirical maximality of LOOD when the query is the differing point: Figure 2a shows a brute-force search over the data domain of a one-dimensional toy sin dataset (as per the setup in Appendix D.1) to find the query with the maximal LOOD and observe that it is indeed the differing data. For the practical CIFAR10 image dataset under NNGP kernels, we examine the LOOD on queries that are perturbations of the differing data. We observe in Figure 2b and the caption that querying on the differing point itself consistently incurs maximal LOOD across 70 runs with random choices of the differing data and random data perturbation directions. Finally, Figure 2c shows information leakage (as captured by membership inference attack performance in AUC score) of NN models on queries near the

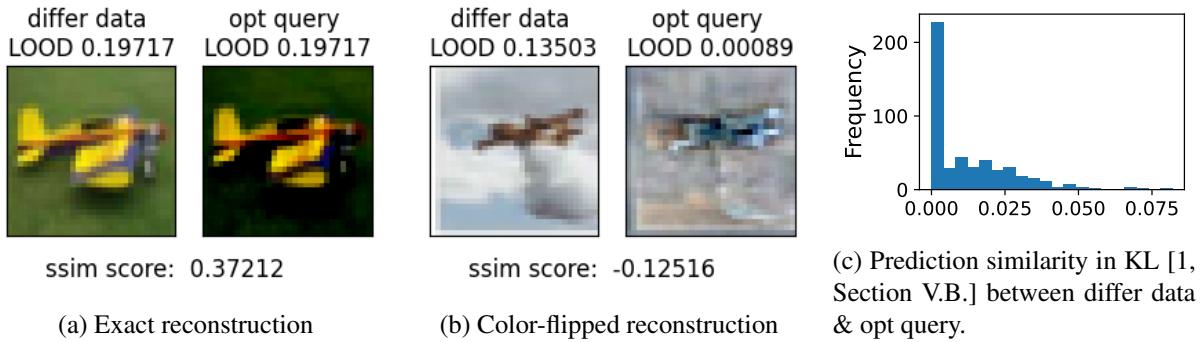


Figure 1: Examples of optimized query after LOOD optimization, given all ‘car’ and ‘plane’ images from CIFAR10 as training dataset, under NNGP for 10-layer FC network with ReLU activation. We show in (a) the optimized query gets close to the differing point in LOOD; and in (b) the optimized query converges to a query with significantly lower LOOD than the differing data, yet they visually resemble each other. In (c) we further statistically check the prediction similarity between query and differing data in KL divergence [1, Section V.B.] across 500 runs with randomly chosen differing data (see Figures 14, 15, 16, 17, 18 and 19 for all the images for these runs).

differing data for five randomly chosen differing data (this was repeated for 70 chosen differing data; see caption); we show that MIA performance is highest when query equals the differing data in 4/5 cases.² Moreover, we show that the empirical maximality of LOOD when query equals the differing data extends to the setting with multiple queries (Appendix E.1) and a group of differing data (Appendix E.2).

Optimized query for LOOD is close to the differing point in a majority of cases. In LOOD optimization, we observe that the optimized query could recover the differing data with high visual precision: see Figure 1a for an example and Figure 14-19 for more examples over randomly chosen differing data points.³ The optimized query also has very similar LOOD to the differing data itself consistently across random experiments, as shown in Figure 1c. This also holds when there are more than one differing data (i.e., leave-one-group-out setting); only in one case does the optimized single query have slightly higher LOOD than the differing record that it recovers – see Figure 4 right-most column and Appendix E.2. We remark that, possibly due to the nonconvex structure of LOOD, cases exist where the optimized query has significantly lower LOOD than the differing data (Figure 1b). Noteworthy is that even in this case the query still resembles the shape of the differing data visually. As we will later show this can be leveraged for data reconstruction attacks, even for the setting where there is a group of differing data–Section 4.2 and Appendix E.2.

When the differing point is not the optimal query. In practice data-dependent normalization is often used prior to training. In such a setting the optimized query has a higher LOOD than the differing point itself – see Appendix D.4 for details.

²There is one special case where the differing image is easy-to-predict and incurs low leakage on all queries.

³We also tried other image similarity metrics in the reconstruction attack literature, such as the per-pixel distance ([1, Section V.B] and [3, Section 5.1]), SSIM score [10, Figure 1] and LPIPS metric [1, Section V.B] – but these metrics do not reflect the visual similarity in our setting. This is possibly because the LOOD optimization objective only recovers the highest amount of information about the differing data which need not be exactly the same RGB values. LOOD optimization results in color dampening (Figure 1a) or color-flipped reconstruction (Figure 1b) of the differing data, which are not reflected by existing metrics such as SSIM score. It is a long-standing open problem to design better image similarity metrics for such reconstructions that do not preserve the exact RGB values.

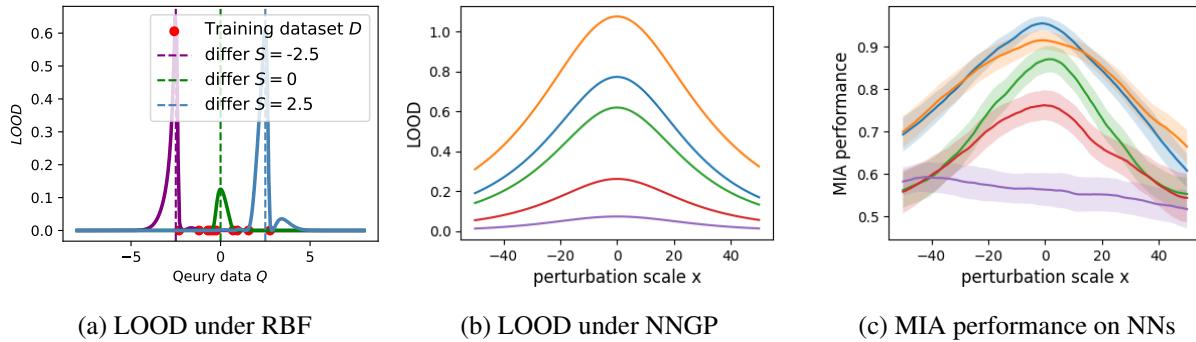


Figure 2: Empirical maximality of information leakage when query equals differing data. Figure (a) uses a toy dataset $D = (x, y)$ with $x \in [-5, 5]$, $y = \sin(x)$, and computes LOOD for all queries in the data domain. Figure (b), (c) use ‘car’ and ‘plane’ images in CIFAR-10, and evaluate leakage on perturbation queries $\{Q = S + x \cdot R : x \in [-50, 50]\}$ of the differing data S along a random direction $R \in L_\infty$ unit ball. For five randomly chosen differing data, we show in (b) and (c) that LOOD and MIA performance are visually maximal around $x = 0$. For statistical pattern, we repeat the experiment for 70 randomly chosen differing data, and observe the scale of perturbation with maximal information leakage is $x = 0.000 \pm 0.000$ for NNGPs, and $x = -2.375 \pm 5.978$ for NNs.

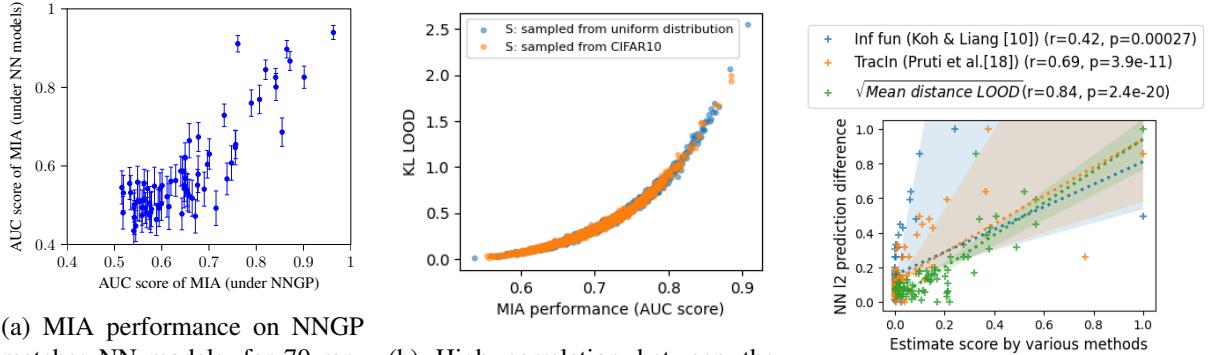
4 LOOD vs. Leakage, Memorization, and Influence Measures

In this section, we show how LOOD can be applied to established definitions of information leakage based on inference attacks, influence functions, and memorization.

4.1 Quantifying Information Leakage using Leave-one-out Distinguishability

Information Leakage relates to the extent to which a model’s output can unintentionally reveal information about the training dataset when the model’s prediction (distribution) at a query Q is observed by an adversary. One established method for assessing it is through a *membership inference attack* (MIA) [30], where an attacker attempts to guess whether a specific data point S was part of the model’s training dataset; the leakage is quantified by the attacker’s false positive rate (FPR) versus his true positive rate (TPR), as well as the area under the TPR-FPR curve (AUC), when performed on randomly trained target models and its training/test data records. However, performing computing the leakage using MIA can be computationally expensive due to the associated costs of retraining many models on a fixed pair of leave-one-out datasets. Using LOOD with Gaussian processes is an efficient alternative that allows to answer: i) Which query Q should the attacker use to extract the highest possible amount of information leakage regarding S ? (Section 3), and ii) How much is a specific record S at risk based on its worst-case query, effectively quantifying the information leakage associated with each training record.

LOOD demonstrates information leakage in NNGP. We first investigate whether the LOOD when querying the differing record itself accurately reflects the level of information leakage in a membership inference attack. For an MIA to successfully distinguish between two prediction distributions of models trained on leave-one-out datasets D and D' with the differing record S , it must be able to effectively differentiate between $f|_{D,\sigma^2}(S)$ and $f|_{D',\sigma^2}(S)$. Intuitively, the success of MIA (as measured by its AUC score) implies a high distinguishability between these two distributions. In Figure 3b, we observe a strong correlation between LOOD and the performance of MIA. This finding leads us to conclude that LOOD effectively reflects information leakage in NNGP predictions.



(a) MIA performance on NNGP

matches NN models, for 70 random training data records, each represented by a dot. For MIA on NNs, we evaluate over 50 models on D and 50 models on D' . For MIA on NNGP, we compute the prediction mean and variance of NNGP (on the differing data) given D and D' , and evaluate MIA over 5000 samples from the two Gaussian distributions. Models are trained on ‘car’ and ‘plane’ images from CIFAR10.

(b) High correlation between the LOOD and MIA performance. The leave-one-out dataset D is a class-balanced subset of CIFAR-10 with size 1000, and $D' = D \cup S$ for a differing record S . We evaluate over 200 randomly chosen S (from CIFAR-10 or uniform distribution over the pixel domain). This indicate that LOOD faithfully captures information leakage under MIA. See Appendix C.3 for more discussions on this correlation curve.

(c) Mean distance LOOD matches the prediction difference after leave-one-out retraining of NN models, for 70 randomly chosen differing data (each represented by a dot). Dataset contains ‘car’ and ‘plane’ images in CIFAR-10. As comparison, we plot the influence function estimates (Koh and Liang [10]) and TracIn scores (Pruthi et al.[18]). We normalize all estimates to $[0, 1]$. We report Pearson’s correlation and p-value in the legends.

Figure 3: Validation of our analytical framework (based on LOOD and mean distance LOOD for NNGP), according to information leakage measure (the performance of membership inference attacks) and influence definition (prediction difference under leave-one-out retraining).

NNGP effectively and efficiently captures leakage under practical neural network training. In Section 3, we establish the theoretical first-order stationary condition, and empirical optimality of LOOD (information leakage) when the query is the differing data point. In this section, we further explore whether the leakage computed using NNGP models accurately reflects the leakage empirically measured in neural network models which are trained using SGD. To measure leakage, we utilize the success of the membership inference attacks (AUC score) when observing predictions from NN models trained on leave-one-out datasets. In Figure 3a, we compare the MIA performance between NNGP models and neural network models. Notably, we observe a strong correlation, suggesting that NNGP captures the information leakage of neural networks about training data records.

One significant advantage of using NNGPs to estimate leakage is the efficiency of analytic expressions, which significantly reduces computational requirements. In the analysis of Figure 3a, we observed over $140\times$ **speed up** for estimating leakage using our framework versus empirically measuring it as it is performed in the literature [2, 35].⁴

4.2 Optimizing LOOD as a Data Reconstruction Attack

A strong threat against machine learning models is an adversary that reconstructs the data points with significant (MIA) likelihood of being in the training set [3]. In Figure 1 and Figure 14-19, we have demonstrated that optimizing LOOD for a query often leads to the recovery of the differing data point.

⁴Launching the empirical MIA in Figure 3a involved training 50×21 FC networks, which took approximately 24 GPU hours. For the same setting, obtaining the LOOD results only required around 10 GPU minutes.

This is due to the significant amount of information that is being leaked through the model’s prediction on the differing data itself. Such results showcase that LOOD can be leveraged for a data reconstruction attack in a setting where the adversary has access to at least part of the train data. Notably, the prior works of [1, 9] focus on reconstruction attacks under the assumption that the adversary possesses knowledge of *all* data points except for one (which aligns with our leave-one-out setting).

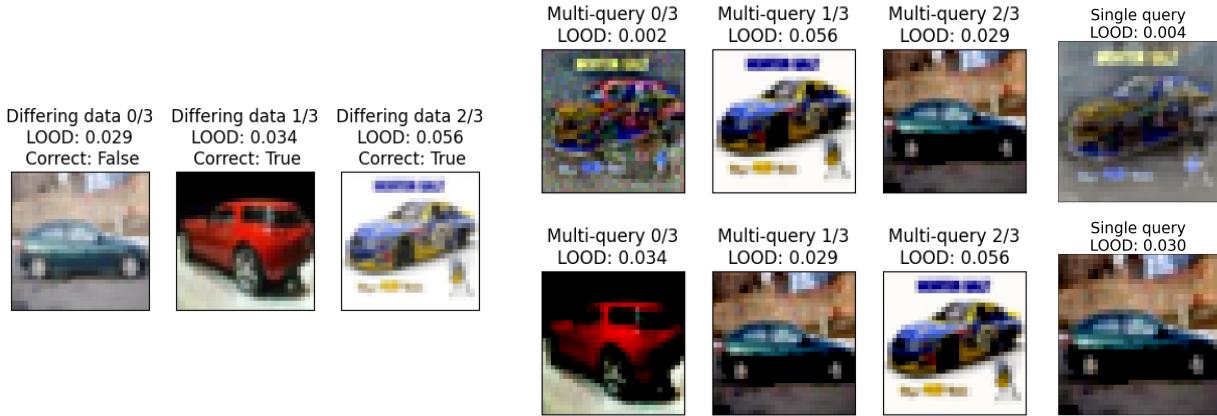


Figure 4: Optimal query (data reconstruction) for leave-one group-out distinguishability. The differing records S are fixed to be a group of three records (as in the three leftmost images in the first row). The three middle images are the optimal queries in the multi-query setting, and the rightmost image is the optimal single query; each row presents the results for a different random initialization. We also show the correctness of trained model’s prediction when removing this data point from the larger training dataset. These examples show that images with highest LOOD and for certain initializations even *all* differing images can be recovered.

Our reconstruction via optimizing LOOD methodology also extends to the group setting. In Figure 4, we present results demonstrating that even when the attacker does not know a *group* of data points and has access to the model distribution on the complete dataset, they can successfully reconstruct the whole group of differing data points with a high degree of visual similarity. Additionally, by examining the frequency of LOOD optimization converging to a query close to each data point in the differing group (across multiple runs and optimized queries), we can assess which S ’s are most vulnerable to reconstruction – see Appendix E.2 for more details. Reconstruction under the leave-one-group-out setting mimics a more practical real world adversary that aims to reconstruct more than one training data. In the limit of large group size that equals the size of the training dataset, such leave-one-group-out setting recovers one of the difficult reconstruction attack scenario studied in [10], where the adversary does not have exact knowledge about any training data. Therefore, extending our results within the leave-one-group-out setting to practical reconstruction attacks presents a promising open problem for future exploration.

4.3 Analysis of Influence and Memorization

Influence functions [27, 16, 15] quantify the degree of change in the model’s prediction (error) for a record Q when a data record S is added to the training dataset D' . A closely related concept is memorization [6, 7], which can be modeled as self-influence. In terms of definitions, certain prior influence [15] and memorization (i.e., self-influence) [6, 7] definitions can be recovered as special cases of LOOD if we measure the distance between the mean predictions of the leave-one-out models on a query, instead

of quantifying the divergence between their distributions. We discuss the detailed relations between the concepts in Appendix F.

Mean distance LOOD under NNGP agrees with leave-one-out NN retraining. In Figure 3c, we show that mean distance LOOD agrees with the actual prediction differences under leave-one-out retraining. In the x -axis, we show the actual prediction difference between 100 NN models trained on each of the leave-one-out datasets, over 50 randomly chosen differing data. This is exactly the memorization or self-influence as defined by [6, Page 5] and [7, Equation 1], and serves as a ground-truth baseline that should be matched by good influence estimations. In the y -axis, we show the normalized mean distance LOOD (under NNGP), influence function [15] and TracIn estimates [26]. We observe that mean distance LOOD has the highest correlation to the actual prediction difference among all methods.

Analysis of Influence and memorization via mean distance LOOD. LOOD can thus be seen as a generalization of the prior metrics that allows to account for higher-order differences in the distributions. Measuring influence and memorization using our framework offers a computationally efficient methodology, and thus allows us to answer questions such as: Where, i.e., at which query Q , does the model exhibit the most influence? Which data record S has the largest influence? How do different model architectures impact influence? Answers to these questions can be derived in a similar manner as the analysis for LOOD that we provide in other parts (Section 3 and Section 5) of the paper. As a concrete example, we show how to use mean distance LOOD for a more fine-grained assessment of influence within our analytical framework. Proposition 4.1 provides valuable insights into influence analysis using the different behaviors of Mean LOOD, based on where the differing point is located with respect to the rest of the training set. This allows us to answer the question: *Which data record S has the most significant influence and at which query Q ?*

Proposition 4.1 (First-order optimality condition for influence). *Consider a dataset D , a differing point S , and a kernel function K satisfying the same conditions of Theorem 3.1. Then, we have that*

$$\nabla_Q M(Q) |_{Q=S} = -\alpha^{-2} (1 - K_{SD} M_D^{-1} K_{DS}) (y_S - K_{SD} M_D^{-1} y_D)^2 \mathring{K}_{SD} M_D^{-1} K_{DS},$$

where $M(Q)$ is the mean distance LOOD in C.1, K_{SD} , K_{DS} , K_{DD} are kernel matrices as defined in Section 2, $\mathring{K}_{SD} = \frac{\partial}{\partial Q} K_{QD} |_{Q=S}$, $M_D = K_{DD} + \sigma^2 I$, and $\alpha = 1 - K_{SD} M_D^{-1} K_{DS} + \sigma^2$.

Our first observation from Proposition 4.1 is that when S is far from the dataset D in the sense that $K_{SD} \approx 0$, the mean distance LOOD gradient $\nabla_Q M(Q) |_{Q=S}$ is close to 0, i.e., the differing data S is a stationary point for the mean distance LOOD objective. We also analyze the second-order Hessian condition of mean distance LOOD in Lemma G.4, and prove that the differing data S has locally maximal influence, as long as S far from the remaining training dataset D , and the data labels are bounded (which is the case in a classification task). In the opposite case where S is close to D in the sense that $y_S \approx K_{SD} M_D^{-1} y_D$ (i.e., S is perfectly predicted by the GP trained on remaining dataset D), then the gradient $\nabla_Q M(Q) |_{Q=S}$ of mean distance LOOD is also close to zero.

Between the above two cases, when the differing data is neither too close nor too far from the remaining dataset, Proposition 4.1 proves that in general $\nabla_Q M(Q) |_{Q=S} \neq 0$, i.e., the differing data S is not optimal for the mean distance LOOD maximization objective Equation (2). To give an intuition, we can think of the Mean LOOD as the result of an interaction of two forces, a force due to the training points, and another force due to the differing point. When the differing point is neither too far nor too close to the training data, the synergy of the two forces creates a region that does not include the differing point where the LOOD is maximal. To further illustrate these patterns and optimality region of mean distance LOOD, we provide numerical examples for RBF kernel on toy dataset in Figures 11b-12, and for NNGPs on image data in Figures 13a-13c in Appendix G.

5 Explaining the Effect of Activation Functions on LOOD

In this section, we theoretically investigate the effect of activation function on the magnitude of LOOD (given fixed leave-one-out datasets). We will leverage Proposition H.1 in the Appendix; it shows that for the same depth, smooth activations such that GeLU are associated with kernels that are farther away from a low rank all-constant matrix (more expressive) than kernel obtained with non-smooth activations, e.g. ReLU. To understand the effect of activation, we only need to analyze how the low-rank property of NNGP kernel matrix affects the magnitude of LOOD (leakage), given a fixed pair of differing point S and remaining dataset D . To this end, we establish the following proposition, which shows that the LOOD (leakage) is small if the NNGP kernel matrix for the training dataset K_{DD} is close to a low rank all-constant matrix.

Proposition 5.1 (Informal: Low rank kernel matrix implies low LOOD). *Let D and $D' = D \cup S$ be an arbitrary pair of leave-one-out datasets, where D contains n records and D' contains $n + 1$ records. Define the function $h(\alpha) = \sup_{x, x' \in D'} |K(x, x') - \alpha|$. Let $\alpha_{\min} = \operatorname{argmin}_{\alpha \in \mathbb{R}^+} h(\alpha)$, and thus $h(\alpha_{\min})$ quantifies how close the kernel matrix is to a low rank all-constant matrix. Assume that $\alpha_{\min} > 0$. Then, there exist constants $A_n, B > 0$ such that*

$$\max_Q \text{LOOD}(f|_{D, \sigma^2}(Q) \| f|_{D', \sigma^2}(Q)) \leq A_n h(\alpha_{\min}) + B n^{-1}.$$

Thus LOOD should decrease as $h(\alpha_{\min})$ decreases, see Appendix H for formal statement and proofs.

Proposition 5.1 proves that the LOOD is upper bounded by the term $h(\alpha_{\min})$ that captures how close the kernel matrix is to an all-constant matrix (of low rank one), thus implying smaller LOOD (leakage) over worst-case query given low rank kernels. Intuitively, this happens because the network output forgets most of the information about the input dataset under low rank kernel matrix.

Analysis: smooth activation functions induce higher LOOD (leakage). We have shown that low rank kernels are associated with smaller LOOD (leakage) over worst-case query (Proposition 5.1) and NNGP kernel converges to a constant kernel at different rates under different activations (Proposition H.1 and [29, 11, 12]). By combining them we conclude that LOOD (leakage) is higher under smooth activations (such as GeLU) than non-smooth activations (such as ReLU). This is consistent with the intuition that smooth activations allow deeper information propagation, thus inducing more expressive kernels and more memorization of the training data. Interestingly, recent work [4] also showed that GeLU enables better model performance than ReLU even in modern architectures such as ViT. Our results thus suggest there is a privacy-accuracy tradeoffs by activation choices.

Empirical results: NNGP with GeLU activation induces higher LOOD than ReLU activation. We now empirically investigate how the activation function affects the LOOD under NNGP. Our experiment setting is NNGP for fully connected neural networks with varying depth $\{2, 4, \dots, 10\}$. The training dataset contains 'plane' and 'car' images from CIFAR10. We evaluated LOOD over 500 randomly chosen differing data, and observe under fixed depth, the computed LOOD under ReLU is more than $1.1 \times$ higher than LOOD under GeLU in 95% cases. We visualize randomly chosen 20 differing data in Figure 5. Under each fixed differing data, we also observe that the gap between LOOD under ReLU and LOOD under GeLU grows with the depth. This is consistent with the pattern predicted by our analysis in Proposition H.1, i.e., GeLU gives rise to higher leakage in LOOD than ReLU activation, and their gap increases with depth. It is worth noting that for our experiments the LOOD optimization is done without constraining the query data norm. Therefore, our experiments shows that the result of Proposition H.1 could be valid in more general settings.

Effect of activation function on LOOD generalizes to information leakage on NN models. We further investigate whether the information leakage (as measured by membership inference attack performance for leave-one-out retrained models) for deep neural networks are similarly affected by activation function. Figure 5 shows that deep neural networks with GeLU activation incur higher per-record information leakage (MIA performance in AUC score) compared to NNs with ReLU activation, for 20 randomly chosen differing data. This is consistent with our analytical results (Proposition H.1 and 5.1) that predict higher LOOD under smooth activation than under ReLU.

6 Conclusions

We have presented a formal framework using Gaussian processes to quantify the influence of individual training data points on predictions of machine learning models. We also explore the collective influence of groups of data points and their potential risk with respect to inference attacks. Our work sheds light on the privacy implications of model predictions on training data, and generalizes the work on influence analysis and memorization.

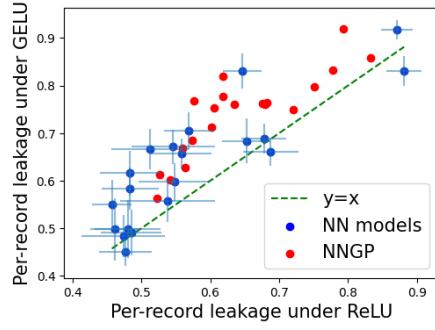


Figure 5: Empirical validation of our analytical results in of Section 5 that per-record information leakage is higher under GELU activation than under ReLU, for both NNGPs and NN models. We evaluate per-record leakage with membership inference attack performance on models trained on leave-one-out datasets. Dataset contains ‘car’ and ‘plane’ images from CIFAR-10. The NN model is fully connected network with depth 10 and width 1024.

References

- [1] Borja Balle, Giovanni Cherubin, and Jamie Hayes. Reconstructing training data with informed adversaries. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1138–1156. IEEE, 2022.
- [2] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022.
- [3] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. *arXiv preprint arXiv:2301.13188*, 2023.
- [4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.
- [5] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.
- [6] Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020.
- [7] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.
- [8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- [9] Chuan Guo, Brian Karrer, Kamalika Chaudhuri, and Laurens van der Maaten. Bounding training data reconstruction in private (deep) learning. In *International Conference on Machine Learning*, pages 8056–8071. PMLR, 2022.
- [10] Niv Haim, Gal Vardi, Gilad Yehudai, Ohad Shamir, and Michal Irani. Reconstructing training data from trained neural networks. *arXiv preprint arXiv:2206.07758*, 2022.
- [11] Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. On the impact of the activation function on deep neural networks training. In *International Conference on Machine Learning*, 2019.
- [12] Soufiane Hayou, Eugenio Clerico, Bobby He, George Deligiannidis, Arnaud Doucet, and Judith Rousseau. Stable resnet. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 1324–1332. PMLR, 13–15 Apr 2021.
- [13] Jiri Hron, Yasaman Bahri, Jascha Sohl-Dickstein, and Roman Novak. Infinite attention: Nnpg and ntk for deep attention networks. In *International Conference on Machine Learning*, pages 4376–4386. PMLR, 2020.

- [14] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private sgd? *Advances in Neural Information Processing Systems*, 33: 22205–22216, 2020.
- [15] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.
- [16] Pang Wei W Koh, Kai-Siang Ang, Hubert Teo, and Percy S Liang. On the accuracy of influence functions for measuring group effects. *Advances in neural information processing systems*, 32, 2019.
- [17] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.
- [18] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.
- [19] Noel Loo, Ramin Hasani, Alexander Amini, and Daniela Rus. Efficient dataset distillation using random feature approximation. *arXiv preprint arXiv:2210.12067*, 2022.
- [20] Yizhang Lou, Chris E Mingard, and Soufiane Hayou. Feature learning and signal propagation in deep neural networks. In *International Conference on Machine Learning*, pages 14248–14282. PMLR, 2022.
- [21] Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlini. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 866–882. IEEE, 2021.
- [22] Milad Nasr, Jamie Hayes, Thomas Steinke, Borja Balle, Florian Tramèr, Matthew Jagielski, Nicholas Carlini, and Andreas Terzis. Tight auditing of differentially private machine learning. *arXiv preprint arXiv:2302.07956*, 2023.
- [23] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.
- [24] Roman Novak, Lechao Xiao, Jaehoon Lee, Yasaman Bahri, Greg Yang, Jiri Hron, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Bayesian deep convolutional networks with many channels are gaussian processes. *arXiv preprint arXiv:1810.05148*, 2018.
- [25] Daniel S. Park, Jaehoon Lee, Daiyi Peng, Yuan Cao, and Jascha Sohl-Dickstein. Towards nngp-guided neural architecture search, 2020.
- [26] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33: 19920–19930, 2020.
- [27] Peter J Rousseeuw, Frank R Hampel, Elvezio M Ronchetti, and Werner A Stahel. *Robust statistics: the approach based on influence functions*. John Wiley & Sons, 2011.
- [28] Samuel S Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. *arXiv preprint arXiv:1611.01232*, 2016.

- [29] S.S. Schoenholz, J. Gilmer, S. Ganguli, and J. Sohl-Dickstein. Deep information propagation. In *International Conference on Learning Representations*, 2017.
- [30] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.
- [31] Yuxin Wen, Arpit Bansal, Hamid Kazemi, Eitan Borgnia, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Canary in a coalmine: Better membership inference with ensembled adversarial queries. *arXiv preprint arXiv:2210.10750*, 2022.
- [32] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.
- [33] Greg Yang. Wide feedforward or recurrent neural networks of any architecture are gaussian processes. *Advances in Neural Information Processing Systems*, 32, 2019.
- [34] Greg Yang and Eta Littwin. Tensor programs ivb: Adaptive optimization in the infinite-width limit. *arXiv preprint arXiv:2308.01814*, 2023.
- [35] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3093–3106, 2022.

Contents

1	Introduction	1
2	Definition and the Analytical Framework	3
3	Optimizing LOOD to Identify the Most Influenced Point	3
4	LOOD vs. Leakage, Memorization, and Influence Measures	6
4.1	Quantifying Information Leakage using Leave-one-out Distinguishability	6
4.2	Optimizing LOOD as a Data Reconstruction Attack	7
4.3	Analysis of Influence and Memorization	8
5	Explaining the Effect of Activation Functions on LOOD	10
6	Conclusions	11
A	Table of Notations	16
B	Additional discussion on the connections between NNGP and NN training	16
C	Deferred details for Section 2	16
C.1	Mean Distance LOOD	16
C.2	KL Divergence LOOD	18
C.3	Comparison between Mean Distance LOOD and KL Divergence LOOD	19
D	Deferred proofs and experiments for Section 3	20
D.1	Deferred proof for Theorem 3.1	20
D.2	Proofs for regularity conditions of commonly used kernel functions	23
D.3	More examples for query optimization under NNGP kernel functions	25
D.4	When the differing point is not the optimal query	25
E	Analyzing LOOD for Multiple Queries and Leave-one-group-out Setting	26
E.1	LOOD under multiple queries	26
E.2	Leave-one-group out distinguishability	26
F	Connections between LOOD and self-influence and memorization	27
G	Optimal query under mean distance and deferred proofs for Section 4.3	28
G.1	Differing point is in general not optimal for Mean Distance LOOD	28
G.2	Differing point is optimal in Mean Distance LOOD when it is far from the dataset	30
G.3	The optimal query for Mean Distance LOOD on image datasets	32
H	Deferred proofs for Section 5	33

A Table of Notations

Table 1: Symbol reference

Symbol	Meaning
D	Leave-one-out dataset
D'	Leave-one-out dataset combined with the differing data
S	Differing data record(s)
s	number of differing data records
Q	Queried data record (s)
q	number of queries
d	dimension of the data
$K(x, x')$	Kernel function
$K(X, Z) \in \mathbb{R}^{p \times k}$, for two finite collections of vectors $X = (x_1, \dots, x_p)$ and $Z = (z_1, \dots, z_k)$	$(K(x_i, z_j))_{\substack{1 \leq i \leq p \\ 1 \leq j \leq k}}$
$K_{XZ} \in \mathbb{R}^{p \times k}$	Abbreviation for $K(X, Z)$
M_D	Abbreviation for $K_{DD} + \sigma^2 \mathbb{I}$
$M_{D'}$	Abbreviation for $K_{D'D'} + \sigma^2 \mathbb{I}$
$M(Q)$	Abbreviation for mean distance LOOD under query Q and leave-one-out datasets D and D'
MIA	Abbreviation for membership inference attack

B Additional discussion on the connections between NNGP and NN training

Note that in general, there is a performance gap between a trained NN and its equivalent NNGP model. However, they are fundamentally connected even when there is a performance gap: the NNGP describes the (geometric) information flow in a randomly initialized NN and therefore captures the early training stages of SGD-trained NNs. In a highly non-convex problem such as NN training, the initialization is crucial and its impact on different quantities (including performance) is well-documented [28]. Moreover, it is also well-known that the bulk of feature learning occurs during the first few epochs of training (see [20, Figure 2]), which makes the initial training stages even more crucial. Hence, it should be expected that properties of NNGP transfer (at least partially) to trained NNs. More advanced tools such as Tensor Programs [34] try to capture the covariance kernel during training, but such dynamics are generally intractable in closed-form.

C Deferred details for Section 2

C.1 Mean Distance LOOD

The simplest way of quantifying the LOOD is by using the distance between the means of the two distributions formed by coupled Gaussian processes:

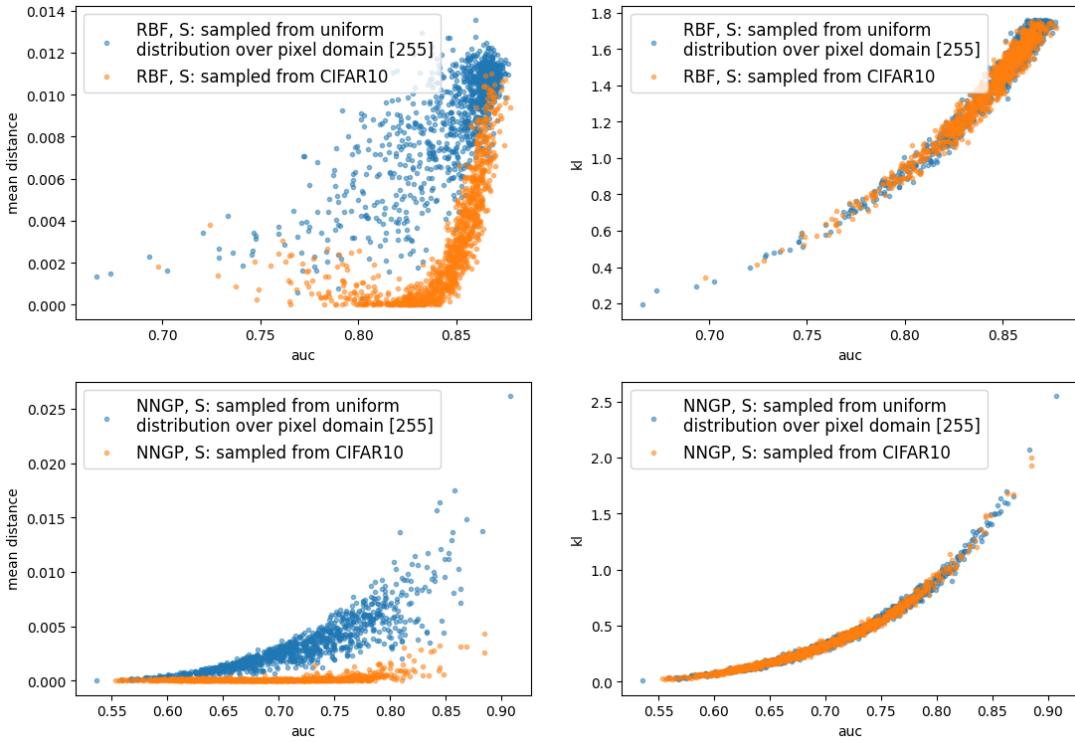


Figure 6: Correlation between the LOOD and MIA success. The leave-one-out dataset D is a class-balanced subset of CIFAR-10 with size 1000, and $D' = D \cup S$ for a randomly chosen differing record S . We evaluate over 200 random choices of the differing data record (sampled from the CIFAR-10 dataset or uniform distribution over pixel domain [255].) We also evaluate under different LOOD metrics (mean distance and KL divergence) and different kernel functions (RBF kernel or NNGP kernel for fully connected network with depth 1). We observe that the correlation between KL and AUC (right figures) is consistently stronger and more consistent than the correlation between mean distance LOOD and auc (left figures).

Definition C.1 (Mean distance LOOD). By computing the distance between the mean of coupled Gaussian processes on query set Q , we obtain the following mean distance LOOD.

$$M(Q; D, D') = \frac{1}{2} \|\mu_{D, \sigma^2}(Q) - \mu_{D', \sigma^2}(Q)\|_2^2 \quad (3)$$

$$= \frac{1}{2} \|K_{QD}(K_{DD} + \sigma^2 \mathbb{I})^{-1} y_D - K_{QD'}(K_{D'D'} + \sigma^2)^{-1} y_{D'}\|_2^2. \quad (4)$$

where Q is a query data record, and $\mu_{D, \sigma^2}(Q)$ and $\mu_{D', \sigma^2}(Q)$ are the mean of the prediction function for GP trained on D and D' respectively, as defined in Section 2. When the context is clear, we abbreviate D and D' in the notation and use $M(Q)$ under differing data S to denote $M(Q; D, D')$ where $D' = D \cup S$.

Example C.2 (Mean distance LOOD is not well-correlated with MIA success). *We show the correlations between Mean distance LOOD and the success of leave-one-out membership inference attack (MIA)⁵ in Figure 6 (left). We observe that low mean distance and high AUC score occur at the same time, i.e., low*

⁵This is measured by the AUC of the attacker’s false positive rate (FPR) versus his true positive rate (TPR).

Mean distance LOOD does not imply low attack success. This occurs especially when the differing point is sampled from a uniform distribution over the pixel domain [0, 255].

Another downside of the mean distance LOOD metric is that solely first-order information is incorporated; this results in it being additive in the queries: $M(Q) = \sum_{i=1}^m M(Q_i)$. That is, Mean distance LOOD does not allow to incorporate the impact of correlations between multiple queries. Moreover, the mean distance LOOD has a less clear interpretation under certain kernel functions. For example, under the NNGP kernel, a query with large ℓ_2 norm will also have high Mean Distance LOOD (despite normalized query being the same), due to the homogeneity of NNGP kernel. By contrast, as we show below, KL divergence (relative entropy) LOOD mitigates these limitations.

C.2 KL Divergence LOOD

To mitigate the downsides of Mean distance LOOD and incorporate higher-order information, we use KL divergence (relative entropy) LOOD.

Definition C.3 (KL Divergence LOOD). Let $Q \in \mathbb{R}^{d \times q}$ be the set of queried data points, where d is the data feature dimension and q is the number of queries. Let D, D' be a pair of leave-one-out datasets such that $D' = D \cup S$ where S is the differing data point. Let $f|_{D,\sigma^2}$ and $f|_{D',\sigma^2}$ as coupled Gaussian processes given D and D' . Then the Leave-one-out KL distinguishability between $f|_{D,\sigma^2}$ and $f|_{D',\sigma^2}$ on queries Q is as follows,

$$KL(f|_{D,\sigma^2}(Q) \| f|_{D',\sigma^2}(Q)) = \frac{1}{2} \left(\log \frac{|\Sigma'(Q)|}{|\Sigma(Q)|} - q + \text{Tr}(\Sigma'(Q)^{-1} \Sigma(Q)) \right) \quad (5)$$

$$+ (\mu(Q) - \mu'(Q))^{\top} \Sigma'(Q)^{-1} (\mu(Q) - \mu'(Q)) \quad (6)$$

where for brevity, we denote $\mu(Q) = \mu_{D,\sigma^2}(Q)$, $\mu'(Q) = \mu_{D',\sigma^2}(Q)$, $\Sigma(Q) = \Sigma_{D,\sigma^2}(Q)$, and $\Sigma'(Q) = \Sigma_{D',\sigma^2}(Q)$ as defined in Section 2.

Example C.4 (KL divergence is better correlated with MIA success). *We show correlations between KL Divergence LOOD and AUC score of the MIA in Figure 6 (right). We observe that KL Divergence LOOD and AUC score are well-correlated with each other.*

On the asymmetry of KL divergence. In this paper we restrict our discussions to the case when the base distribution in KL divergence is specified by the predictions on the larger dataset D' , which is a reasonable upper bound on the KL divergence with base distribution specified by predictions on the smaller dataset. This is due to the following observations:

1. In numerical experiments (e.g., the middle figure in Figure 7) that $\Sigma(Q)/\Sigma'(Q) \geq 1$ across varying single query Q . This is aligned with the intuition that the added information in the larger dataset D reduce its prediction uncertainty (variance).
2. If $\Sigma(Q)/\Sigma'(Q) \geq 1$, then we have that

$$KL(f|_{D,\sigma^2}(Q) \| f|_{D',\sigma^2}(Q)) \geq KL(f|_{D',\sigma^2}(Q) \| f|_{D,\sigma^2}(Q))$$

Proof. Denote $r = \Sigma(Q)/\Sigma'(Q) \geq 1$, then by Equation (6), for single query Q , we have that

$$KL(f|_{D,\sigma^2}(Q) \| f|_{D',\sigma^2}(Q)) = \frac{1}{2} (-\log r - 1 + r + 2M(Q)\Sigma'(Q)^{-1}) \quad (7)$$

where $M(Q)$ is the abbreviation for mean distance LOOD as defined in Definition C.1. Similarly, when the base distribution of KL divergence changes, we have that

$$KL(f|_{D',\sigma^2}(Q)\|f|_{D,\sigma^2}(Q)) = \frac{1}{2} \left(\log r - 1 + \frac{1}{r} + 2M(Q)\Sigma(Q)^{-1} \right) \quad (8)$$

Therefore, we have that

$$\begin{aligned} &KL(f|_{D,\sigma^2}(Q)\|f|_{D',\sigma^2}(Q)) - KL(f|_{D',\sigma^2}(Q)\|f|_{D,\sigma^2}(Q)) \\ &= \frac{1}{2} \left(r - \frac{1}{r} - 2 \log r + 2M(Q)\Sigma(Q)^{-1}(r-1) \right) \end{aligned}$$

For $r \geq 1$, we have that $r - \frac{1}{r} - 2 \log r \geq 0$ and $M(Q)\Sigma(Q)^{-1}(r-1) \geq 1$. Therefore, we have that $KL(f|_{D,\sigma^2}(Q)\|f|_{D',\sigma^2}(Q)) - KL(f|_{D',\sigma^2}(Q)\|f|_{D,\sigma^2}(Q)) \geq 0$ as long as $r = \Sigma(Q)/\Sigma'(Q) \geq 1$. \square

Therefore, the KL divergence with base prediction distribution specified by the larger dataset is an upper bound for the worst-case KL divergence between prediction distributions on leave-one-out datasets.

C.3 Comparison between Mean Distance LOOD and KL Divergence LOOD

KL Divergence LOOD is better correlated with attack performance. As observed from Figure 6, KL divergence LOOD exhibits strong correlation to attack performances in terms of AUC score of the corresponding leave-one-out MIA. By contrast, Mean distance LOOD is not well-correlated with MIA AUC score. For example, there exist points that incur small mean distance LOOD while exhibiting high privacy risk in terms of AUC score under leave-one-out MIA. Moreover, from Figure 6 (right), empirically there exists a general threshold of KL Divergence LOOD that implies small AUC (that persists under different choices of kernel function, and dataset distribution). This is reasonable since KL divergence is the mean of negative log-likelihood ratio, and is thus closely related to the power of log-likelihood ratio membership inference attack.

Why the mean is (not) informative for leakage. To understand when and why there are discrepancies between the mean distance LOOD and KL LOOD, we numerically investigate the optimal query Q^* that incurs maximal mean distance LOOD. In Figure 7, we observe that although Q^* indeed incurs the largest mean distance among all possible query data, the ratio $\log(\Sigma(Q)/\Sigma'(Q))$ between prediction variance of GP trained on D and that of GP trained on D' , is also significantly smaller at query Q^* when compared to the differing point S . Consequently, after incorporating the second-order variance information, the KL divergence at query Q^* is significantly lower than the KL divergence at the differing point S^* , despite of the fact that Q^* incurs maximal mean distance LOOD. A closer look at Figure 7 middle indicates that the variance ratio peaks at the differing point S^* , and drops rapidly as the query becomes farther away from the differing data and the training dataset. Intuitively, this is because $\Sigma'(Q)$, i.e., the variance at the larger dataset, is extremely small at the differing data point S . As the query gets further away from the larger training dataset, $\Sigma'(Q)$ then increases rapidly. By contrast, the variance on the smaller dataset $\Sigma(Q)$ is large at the differing data point S , but does not change a lot around its neighborhood, thus contributing to low variance ratio. This shows that there exist settings where the second order information contributes significantly to the information leakage, which is not captured by mean distance LOOD (as it only incorporates first-order information).

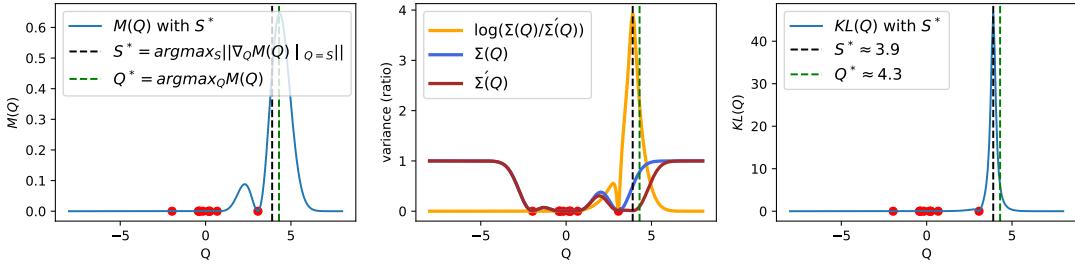


Figure 7: An example on a one-dimensional toy sine training dataset generated as Appendix G.1, where each training data is shown as a red dot. We denote S^* as a crafted differing data record that incurs maximum gradient for the mean distance LOOD objective, which is constructed following the instructions in Algorithm 1. In the left plot, we observe that the optimal query Q^* for mean distance LOOD does not equal the differing point S^* . On the contrary, the optimal query for KL LOOD (which incurs maximal information leakage) is the differing point S^* (right plot). This shows the discrepancy between mean distance LOOD and KL LOOD, and shows the inadequacy of using mean distance LOOD to capture information leakage.

As another example case of when the *variance of prediction distribution is non-negligible* for faithfully capturing information leakage, we investigate under kernel functions that are homogeneous with regard to their input data. Specifically, for a homogeneous kernel function K that satisfies $K(\lambda x, x') = K(x, \lambda x') = \lambda^\alpha K(x, x')$ for any real number $\lambda \in \mathbb{R}$, we have that

$$\begin{aligned}\Sigma_{D,\sigma^2}(\lambda Q) &= \lambda^{2\alpha} \Sigma_{D,\sigma^2}(Q) \\ \mu_{D,\sigma^2}(\lambda Q) &= \lambda^\alpha \mu_{D,\sigma^2}(Q),\end{aligned}$$

where $\sigma^2 > 0$ denotes the variance of the label noise. Therefore, for any $\lambda \in \mathbb{R}$, we have that

$$M(\lambda Q; D, D') = \lambda^\alpha M(Q; D, D') \tag{9}$$

$$KL(f|_{D,\sigma^2}(\lambda Q) \| f|_{D',\sigma^2}(\lambda Q)) = KL(f|_{D,\sigma^2}(Q) \| f|_{D',\sigma^2}(Q)) \tag{10}$$

Consequently, the query that maximizes Mean distance LOOD explodes to infinity for data with unbounded domain. On the contrary, the KL Divergence LOOD is scale-invariant and remains stable as the query scales linearly (because it takes the variance into account, which also grows with the query scale). In this case, the mean distance is not indicative of the actual information leakage as the variance is not a constant and is non-negligible.

In summary, KL LOOD is a metric that is better-suited for the purpose of measuring information leakage, compared to mean distance LOOD. This is because KL LOOD captures the correlation between multiple queries, and incorporates the second-order information of prediction distribution.

D Deferred proofs and experiments for Section 3

D.1 Deferred proof for Theorem 3.1

In this section, we prove that the differing data record is the stationary point for optimizing the LOOD objective. We will need the following lemma.

Lemma D.1. For $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$ and $c \in \mathbb{R}$, we have that

$$\begin{pmatrix} A & b \\ b^\top & c \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} + \alpha^{-1} A^{-1} b b^\top A^{-1} & -\alpha^{-1} A^{-1} b \\ -\alpha^{-1} b^\top A^{-1} & \alpha^{-1} \end{pmatrix} \quad (11)$$

where $\alpha = c - b^\top A^{-1} b$.

We now proceed to prove Theorem 3.1

Theorem 3.1. Let the kernel function be K . Assume that (i) $K(x, x) = 1$ for all training data x , i.e., all training data are normalized under kernel function K ; (ii) $\frac{\partial K(x, x')}{\partial x} |_{x'=x} = 0$ for all training data x , i.e., the kernel distance between two data records x and x' is stable when they are equal. Then, the differing data is a stationary point for LOOD as a function of query data, as follows

$$\nabla_Q \text{LOOD}(f|_{D, \sigma^2}(Q) \| f|_{D', \sigma^2}(Q))|_{Q=S} = 0.$$

Proof. In this statement, LOOD refers to KL LOOD Definition C.3, therefore by analytical differentiation of Definition C.3, we have that

$$\frac{\partial}{\partial Q} \text{LOOD}(f|_{D, \sigma^2}(Q) \| f|_{D', \sigma^2}(Q)) = \frac{\partial}{\partial Q} \text{KL}(f|_{D, \sigma^2}(Q) \| f|_{D', \sigma^2}(Q)) \quad (12)$$

$$\begin{aligned} &= \left(1 - \frac{\Sigma'(Q)}{\Sigma(Q)}\right) \cdot \frac{\partial}{\partial Q} \left(\frac{\Sigma(Q)}{\Sigma'(Q)}\right) \\ &\quad + \frac{\partial \|\mu(Q) - \mu'(Q)\|_2^2}{\partial Q} \Sigma'(Q)^{-1} \\ &\quad + \|\mu(Q) - \mu'(Q)\|_2^2 \frac{\partial \Sigma'(Q)^{-1}}{\partial Q}, \end{aligned} \quad (13)$$

Note that here $\Sigma(Q)$ and $\Sigma'(Q)$ are both scalars, as we consider single query. We denote the three terms by $F_1(Q)$, $F_2(Q)$, and $F_3(Q)$. We will show that $F_1(S) = 0$ and $F_2(S) = -F_3(S)$ which is sufficient to conclude. Let us start with the first term.

1. $F_1(S) = 0$: Recall from the Gaussian process definition in Section 2 that

$$\Sigma(Q) = K_{QQ} - K_{QD} M_D^{-1} K_{DQ}.$$

where $M_D = K_{DD} + \sigma^2 I$. Therefore,

$$\frac{\partial}{\partial Q} \Sigma(Q) = -2 \frac{\partial K_{QD}}{\partial Q} M_D^{-1} K_{DQ},$$

where $\frac{\partial K_{QD}}{\partial Q} \in \mathbb{R}^{d \times n}$ (d is the dimension of Q and n is the number of datapoints in the training set D). A similar formula holds for $\frac{\partial}{\partial Q} \Sigma'(Q)$.

From Lemma D.1, and by using the kernel assumption that $K(S, S) = 1$, we have that

$$M_{D'}^{-1} = \begin{pmatrix} K_{DD} + \sigma^2 I & K_{DS} \\ K_{SD} & 1 + \sigma^2 \end{pmatrix}^{-1} \quad (14)$$

$$= \begin{pmatrix} M_D^{-1} + \alpha^{-1} M_D^{-1} K_{DS} K_{SD} M_D^{-1} & -\alpha^{-1} M_D^{-1} K_{DS} \\ -\alpha^{-1} K_{SD} M_D^{-1} & \alpha^{-1} \end{pmatrix} \quad (15)$$

where $\alpha = 1 + \sigma^2 - K_{SD}M_D^{-1}K_{DS} = \sigma^2 + \Sigma(S)$.

Using the fact that $M_{D'}e_{n+1} = K_{D'S} + \sigma^2 e_{n+1}$ where $e_{n+1} = (0, \dots, 0, 1)^\top \in \mathbb{R}^{n+1}$, simple calculations yield

$$\begin{aligned}\Sigma'(S) &= 1 - K_{SD'}M_{D'}^{-1}K_{D'S} \\ &= \sigma^2(1 - \sigma^2 e_{n+1}^\top M_{D'}^{-1} e_{n+1}) \\ &= \sigma^2(1 - \sigma^2 \alpha^{-1}).\end{aligned}$$

We now have all the ingredients for the first term. To alleviate the notation, we denote by $\mathring{K}_{SD} = \frac{\partial}{\partial Q} K_{QD} |_{Q=S}$.

We obtain

$$\frac{\partial}{\partial Q} \left(\frac{\Sigma(Q)}{\Sigma'(Q)} \right) |_{Q=S} = \frac{-2\mathring{K}_{SD}M_D^{-1}K_{DS}}{\Sigma'(S)} + \frac{2\Sigma(S)\mathring{K}_{SD'}M_{D'}^{-1}K_{D'S}}{\Sigma'(S)^2}.$$

A key observation here is the fact that the last entry of $\mathring{K}_{SD'}$ is 0 by assumption. Using the formula above for $M_{D'}^{-1}$, we obtain

$$\begin{aligned}\mathring{K}_{SD'}M_{D'}^{-1}K_{D'S} &= -\sigma^2 \mathring{K}_{SD'}M_{D'}^{-1}e_{n+1} \\ &= -\sigma^2 \mathring{K}_{SD}(-\alpha^{-1}M_D^{-1}K_{DS}) \\ &= \alpha^{-1}\sigma^2 \mathring{K}_{SD}M_D^{-1}K_{DS}.\end{aligned}$$

Now observe that $\alpha^{-1}\sigma^2\Sigma(S)\Sigma'(S)^{-1} = \alpha^{-1}\sigma^2(\alpha - \sigma^2)\sigma^{-2}(1 - \sigma^2\alpha^{-1})^{-1} = 1$, which concludes the proof for the first term.

2. $F_2(S) + F_3(S) = 0$:

Let us start by simplifying $F_2(S) = \frac{\partial \|\mu(Q) - \mu'(Q)\|_2^2}{\partial Q} |_{Q=S} \Sigma'(S)^{-1}$. The derivative here is that of $M(Q)$, and we have

$$\frac{\partial \|\mu(Q) - \mu'(Q)\|_2^2}{\partial Q} |_{Q=S} = 2(\mu(S) - \mu'(S))(\mathring{K}_{SD}M_D^{-1}y_D - \mathring{K}_{SD'}M_{D'}^{-1}y_{D'}).$$

Using the formula of $M_{D'}^{-1}$ and observe that the last entry of \mathring{K}_{SD} is zero, we obtain

$$\frac{\partial \|\mu(Q) - \mu'(Q)\|_2^2}{\partial Q} |_{Q=S} = 2\alpha^{-1}(\mu(S) - \mu'(S))(y_S - K_{SD}M_D^{-1}y_D)\mathring{K}_{SD}M_D^{-1}K_{DS}.$$

Moreover, we have that

$$\mu(S) - \mu'(S) = K_{SD}M_D^{-1}y_D - K_{SD'}M_{D'}^{-1}y_{D'} \tag{16}$$

$$= K_{SD}M_D^{-1}y_D - e_{n+1}^\top(M_{D'} - \sigma^2 I)M_{D'}^{-1}y_{D'} \tag{17}$$

$$= K_{SD}M_D^{-1}y_D - y_S + \sigma^2 e_{n+1}^\top M_{D'}^{-1}y_{D'} \tag{18}$$

$$= K_{SD}M_D^{-1}y_D - y_S + \sigma^2(-\alpha^{-1}K_{SD}M_D^{-1}y_D + \alpha^{-1}y_S) \tag{19}$$

$$= (\sigma^2\alpha^{-1} - 1)(y_S - K_{SD}M_D^{-1}y_D). \tag{20}$$

Therefore

$$F_2(S) = 2\alpha^{-1}(\sigma^2\alpha^{-1}-1)\Sigma'(S)^{-1}(y_S - K_{SD}M_D^{-1}y_D)^2 \mathring{K}_{SD}M_D^{-1}K_{DS} \quad (21)$$

Let us now deal with $F_3(S) = (\mu(S) - \mu'(S))^2 \frac{\partial \Sigma'(Q)^{-1}}{\partial Q} |_{Q=S}$. We have that

$$\begin{aligned} \frac{\partial \Sigma'(Q)^{-1}}{\partial Q} |_{Q=S} &= 2\Sigma'(S)^{-2}\mathring{K}_{SD'}M_{D'}^{-1}K_{D'S} \\ &= -2\sigma^2\Sigma'(S)^{-2}\mathring{K}_{SD'}M_{D'}^{-1}e_{n+1} \\ &= -2\sigma^2\Sigma'(S)^{-2}\mathring{K}_{SD}(-\alpha^{-1}M_D^{-1}K_{DS}) \\ &= 2\alpha^{-1}\sigma^2\Sigma'(S)^{-2}\mathring{K}_{SD}M_D^{-1}K_{DS}. \end{aligned} \quad (22)$$

By plugging Equation (22) and 20 into $F_3(S) = (\mu(S) - \mu'(S))^2 \frac{\partial \Sigma'(Q)^{-1}}{\partial Q} |_{Q=S}$, we prove that

$$F_3(S) = 2\alpha^{-1}\sigma^2(\sigma^2\alpha^{-1}-1)^2\Sigma'(S)^{-2}(y_S - K_{SD}M_D^{-1}y_D)^2 \mathring{K}_{SD}M_D^{-1}K_{DS}.$$

We conclude by observing that $\sigma^2(\sigma^2\alpha^{-1}-1)\Sigma'(S)^{-1} = \sigma^2(\sigma^2\alpha^{-1}-1)\sigma^{-2}(1-\sigma^2\alpha^{-1})^{-1} = -1$.

□

D.2 Proofs for regularity conditions of commonly used kernel functions

We first show that both the isotropic kernel and the correlation kernel satisfies the condition of Theorem 3.1. We then show that the RBF kernel, resp. the NNGP kernel on a sphere, is an isotropic kernel, resp. a correlation kernel.

Proposition D.2 (Isotropic kernels satisfy conditions in Theorem 3.1). *Assume that the kernel function K is isotropic, i.e. there exists a continuously differentiable function g such that $K(x, y) = g(\|x - y\|)$ for all x, y . Assume that g satisfies $z^{-1}g'(z)$ is bounded⁶ and $g(0) = 1$. Then, K satisfies the conditions of Theorem 3.1.*

Proof. It is straightforward that $K(x, x) = 1$ for all x . Simple calculations yield

$$\frac{\partial K(x, y)}{\partial x} = \frac{g'(\|x - y\|)}{\|x - y\|}(x - y).$$

By assumption, the term $\frac{g'(\|x-y\|)}{\|x-y\|}$ is bounded for all $x \neq y$ and also in the limit $x \rightarrow y$. Therefore, it is easy to see that

$$\frac{\partial K(x, y)}{\partial x} |_{x=y} = 0.$$

More rigorously, the partial derivative above exists by continuation of the function $z \rightarrow \frac{g'(z)}{z}z$ near 0^+ . □

Proposition D.3 (Correlation kernels satisfy conditions in Theorem 3.1). *Assume that the kernel function K is a correlation kernel, i.e. there exists a function g such that $K(x, y) = g\left(\frac{\langle x, y \rangle}{\|x\|\|y\|}\right)$ for all $x, y \neq 0$. Assume that $g(1) = 1$. Then, K satisfies the conditions of Theorem 3.1.*

⁶Actually, only the boundedness near zero is needed.

Proof. The first condition is satisfied by assumption on g . For the second condition, observe that

$$\frac{\partial K(x, y)}{\partial x} = \frac{1}{\|x\|\|y\|} \left(y - \frac{\langle x, y \rangle}{\|x\|^2} x \right) g' \left(\frac{\langle x, y \rangle}{\|x\|\|y\|} \right),$$

which yields $\frac{\partial K(x, y)}{\partial x} |_{x=y} = 0$. \square

We have the following result for the NNGP kernel.

Proposition D.4 (NNGP kernel on the sphere satisfies conditions in Theorem 3.1). *The NNGP kernel function is similar to the correlation kernel. For ReLU activation function, it has the form*

$$K(x, y) = \|x\|\|y\|g \left(\frac{\langle x, y \rangle}{\|x\|\|y\|} \right).$$

This kernel function does not satisfy the conditions of Theorem 3.1 and therefore we cannot conclude that the differing point is a stationary point. Besides, in practice, NNGP is used with normalized data in order to avoid any numerical issues; the datapoints are normalized to have $\|x\| = r$ for some fixed $r > 0$. With this formulation, we can actually show that the kernel K restricted to the sphere satisfies the conditions of Theorem 3.1. However, as one might guess, it is not straightforward to compute a "derivative on the sphere" as this requires a generalized definition of the derivative on manifolds (the coordinates are not free). To avoid dealing with unnecessary complications, we can bypass this problem by considering the spherical coordinates instead. For $x \in \mathbb{R}^d$, there exists a unique set of parameters $(r, \varphi_1, \varphi_2, \dots, \varphi_{d-1}) \in \mathbb{R}^d$ s.t.

$$\begin{cases} x_1 = r \cos(\varphi_1) \\ x_2 = r \sin(\varphi_1) \cos(\varphi_2) \\ x_3 = r \sin(\varphi_1) \sin(\varphi_2) \cos(\varphi_3) \\ \vdots \\ x_{d-1} = r \sin(\varphi_1) \sin(\varphi_2) \dots \sin(\varphi_{d-2}) \cos(\varphi_{d-1}) \\ x_d = r \sin(\varphi_1) \sin(\varphi_2) \dots \sin(\varphi_{d-2}) \sin(\varphi_{d-1}). \end{cases}$$

This is a generalization of the polar coordinates in 2D.

Without loss of generality, let us assume that $r = 1$ and let $\varphi = (\varphi_1, \varphi_2, \dots, \varphi_{d-1})$. In this case, the NNGP kernel, evaluated at (φ, φ') is given by

$$K(\varphi, \varphi') = g \left(\sum_{i=1}^{d-1} \left[\prod_{j=1}^{i-1} \sin(\varphi_j) \sin(\varphi'_j) \right] \cos(\varphi_i) \cos(\varphi'_i) + \prod_{j=1}^{d-1} \sin(\varphi_j) \sin(\varphi'_j) \right),$$

with the convention $\prod_{i=1}^0 \cdot = 1$.

Consider the spherical coordinates introduced above, and assume that $r = 1$. Then, the NNGP kernel given by $K(\varphi, \varphi')$ satisfies the conditions of Theorem 3.1.

Proof. Computing the derivative with respect to φ is equivalent to computing the derivative on the unit sphere. For $l \in \{1, \dots, d-1\}$, we have

$$\frac{\partial K(\varphi, \varphi')}{\partial \varphi_l} = J(\varphi, \varphi') \times g' \left(\sum_{i=1}^{d-1} \left[\prod_{j=1}^{i-1} \sin(\varphi_j) \sin(\varphi'_j) \right] \cos(\varphi_i) \cos(\varphi'_i) + \prod_{j=1}^{d-1} \sin(\varphi_j) \sin(\varphi'_j) \right),$$

where

$$\begin{aligned}
J(\varphi, \varphi') = & \sum_{i=l+1}^{d-1} \left[\cos(\varphi_l) \prod_{\substack{j=1 \\ j \neq l}}^{i-1} \sin(\varphi_j) \right] \left[\prod_{j=1}^{i-1} \sin(\varphi'_j) \right] \cos(\varphi_i) \cos(\varphi'_i) \\
& + \cos(\varphi_l) \left[\prod_{\substack{j=1 \\ j \neq l}}^{d-1} \sin(\varphi_j) \right] \left[\prod_{j=1}^{d-1} \sin(\varphi'_j) \right] - \left[\prod_{j=1}^{l-1} \sin(\varphi_j) \right] \left[\prod_{j=1}^{l-1} \sin(\varphi'_j) \right] \sin(\varphi_l) \cos(\varphi'_l).
\end{aligned}$$

When $\varphi = \varphi'$, we obtain

$$\begin{aligned}
J(\varphi, \varphi) = & \frac{\cos(\varphi_l)}{\sin(\varphi_l)} \sum_{i=l+1}^{d-1} \left[\prod_{j=1}^{i-1} \sin(\varphi_j)^2 \right] \cos(\varphi_i)^2 + \frac{\cos(\varphi_l)}{\sin(\varphi_l)} \left[\prod_{j=1}^{d-1} \sin(\varphi_j)^2 \right] \\
& - \frac{\cos(\varphi_l)}{\sin(\varphi_l)} \left[\prod_{j=1}^l \sin(\varphi_j)^2 \right].
\end{aligned}$$

By observing that

$$\sum_{i=l+1}^{d-1} \left[\prod_{j=1}^{i-1} \sin(\varphi_j)^2 \right] \cos(\varphi_i)^2 + \left[\prod_{\substack{j=1 \\ j \neq l}}^{d-1} \sin(\varphi_j)^2 \right] = \prod_{j=1}^l \sin(\varphi_j)^2,$$

we conclude that $\frac{\partial K(\varphi, \varphi')}{\partial \varphi_l} |_{\varphi=\varphi'} = 0$ for all l . □

D.3 More examples for query optimization under NNGP kernel functions

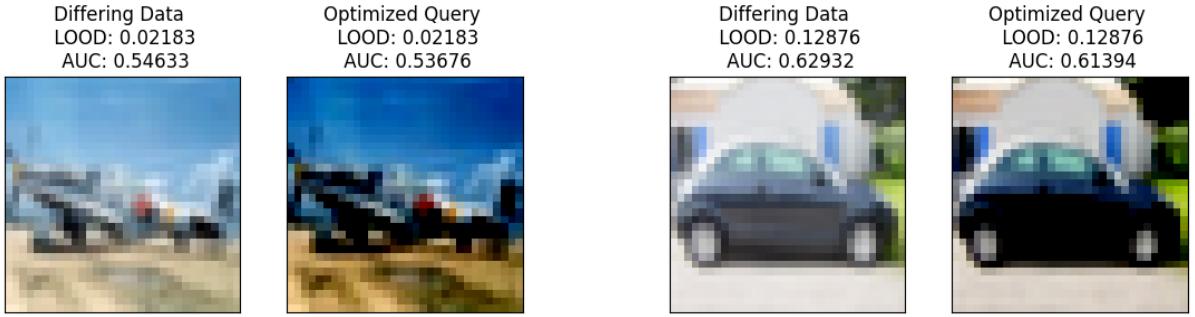
We show the optimized query for NNGP kernel under different model architectures, including fully connected network and convolutional neural network with various depth, in Figure 8.

To further understand the query optimization process in details, we visualize an example LOOD optimization process below in Figure 9.

D.4 When the differing point is not the optimal query

In practice data-dependent normalization is used prior to training, such as batch and layer normalization. Note that this normalization of the pair of leave-one-out datasets results in them differing in more than one record⁷ Then the optimised query data record obtains a higher LOOD than the LOOD of the differing points; we show this in Figure 10; thus in such a setting there is value for the attacker to optimise the query; this is consistent with results from [31] that exploit the training data augmentation.

⁷During inference, the same test record will again go through a normalization step that depends on the training dataset before being passed as a query the corresponding Gaussian process.



(a) A run of LOOD optimization under NNGP kernel for fully connected network with depth 5

(b) A run of LOOD optimization under NNGP kernel for CNN with depth 5

Figure 8: Additional LOOD optimization results under NNGP kernels with different architectures. We evaluate on class-balanced subset of CIFAR-10 dataset of size 1000. The optimized query consistently has similar LOOD and MIA AUC score to the differing data record.

E Analyzing LOOD for Multiple Queries and Leave-one-group-out Setting

E.1 LOOD under multiple queries

We now turn to a more general question of quantifying the LOOD when having a model answer **multiple queries** $\{Q_1, \dots, Q_q\}$ rather than just one query Q (this corresponds to the practical setting of e.g. having API access to a model that one can query multiple times). In this setting, a natural question is: as the number of queries grows, does the distinguishability between leave-one-out models’ predictions get stronger (when compared to the single query setting)? Our empirical observations indicate that the significant information regarding the differing data point resides within its prediction on that data record. Notably, the optimal single query $Q = S$ exhibits a higher or equal degree of leakage compared to any other query set.

Specifically, to show that the optimal single query $Q = S$ exhibits a higher or equal degree of leakage compared to any other query set, we performed multi-query optimization $\max_{Q_1, \dots, Q_q} LOOD(Q_1, \dots, Q_q; D, D')$, which uses gradient descent to optimize the LOOD across q queries, for varying numbers of queries $q = 1, 2, 4, 8$. We evaluated this setting under the following conditions: NNGP using a fully connected network with depths of 2, 4, 6, 8, 10; using 500 random selections of leave-one-out datasets (where the larger dataset is of size 10000) sampled from the ‘car’ and ‘plane’ classes of the CIFAR10 dataset. Our observations indicate that, in all evaluation trials, the LOOD values obtained from the optimized m queries are lower than the LOOD value from the single differing data record.

E.2 Leave-one-group out distinguishability

Here we extend our analysis to the **leave-one group-out** setting, which involves two datasets, D and D' , that differ in a *group* of s data points $\{S_1, \dots, S_s\}$. There are no specific restrictions on how these points are selected. In this setup, we evaluate the model using queries $\{Q_1, \dots, Q_q\}$, where $q \geq 1$. The leave-one group-out approach can provide insights into privacy concerns when adding a group to the training set, such as in datasets containing records from members of the same organization or family [5]. Additionally, this setting can shed light on the success of data poisoning attacks, where an adversary

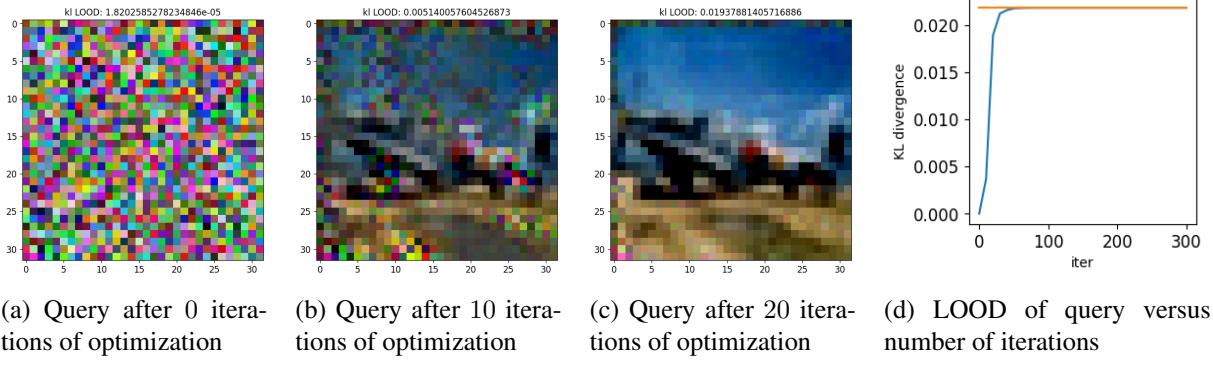


Figure 9: Illustration of an example LOOD optimization process at different number of optimization iterations. We evaluate on a class-balanced subset of CIFAR-10 dataset of size 1000 and NNGP kernel for fully connected network with depth 5. The query is initialized by uniform sampling from the pixel domain, and starts recovering the differing data record as the optimization proceeds. We plot the optimized queries after 0, 10 and 20 iterations and show the convergence of LOOD for query with regard to the increasing number of iterations for LOOD optimization.

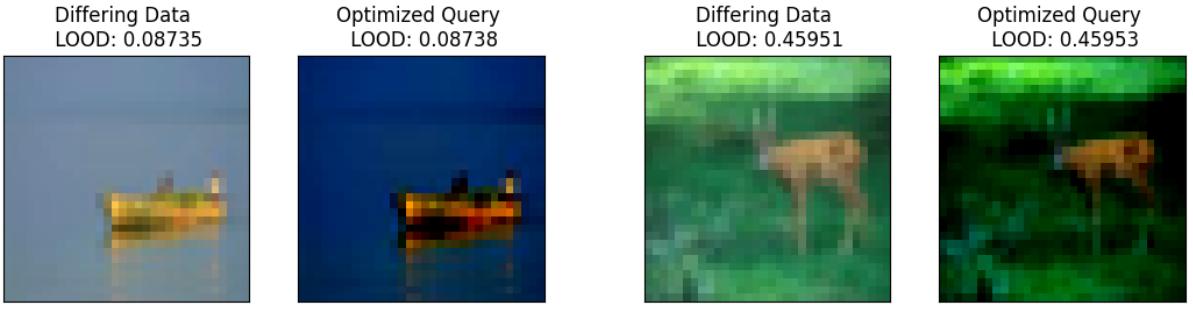
adds a set of points to the training set to manipulate the model or induce wrong predictions. Here, we are interested in constructing and studying queries that result in the highest information leakage about the group, i.e., where the model’s predictions are most affected under the leave-one group-out setting.

Similar to our previous approach, we run maximization of LOOD to identify the queries that optimally distinguishes the leave-one group-out models. Figure 4 presents the results for single-query ($q = 1$) setting and multi-query ($q = 3$) setting, with a differing group size $|S| = 3$. We have conducted in total 10 rounds of experiments, where in each round we randomly select a fixed group of differing datapoints and then perform 10 independent optimization runs (starting from different initializations). Among all the runs, 80% of them converged, in which we observe that the optimal queries are visually similar to the members of the differing group, but the chance of recovering them varies based on how much each individual data point is exposed through a single query (by measuring LOOD when querying the point). We observe that among the points in set S , the differing data point with the highest single-query LOOD was recovered 61% of the times, whereas the one with the lowest single-query LOOD was recovered only 8% of the times. This can be leveraged for data reconstruction attacks where the adversary has access to part of the train dataset of the model.

F Connections between LOOD and self-influence and memorization

Here we provide details on the connections between LOOD and self-influence and memorization.

Connections among mean distance LOOD and self-influence. The work of [15] aims to compute $L(\hat{\theta}_{-z}; z_{test}) - L(\hat{\theta}; z_{test})$, where $\hat{\theta}$ is the minimizer of the training loss, $\hat{\theta}_{-z}$ is the minimizer of loss on all training data excluding record z , and z_{test} is a test data record. This quantity is thus the change of a trained model’s loss on the test point z_{test} due to the removal of one point z from the training dataset (when $z = z_{test}$, it becomes self-influence). We note that this is exactly mean distance LOOD (Definition B.1), given the idealized training algorithm that deterministically outputs the optimal model $\hat{\theta}$ for any training dataset. [15] then use influence functions based on (non-exact) Hessian computations to approximate the above change (as exact Hessian computation is computationally expensive for large



(a) Differing data and optimized query in run one

(b) Differing data and optimized query in run two

Figure 10: Example runs of LOOD optimization under data-dependent training data normalization (that uses per-channel mean and standard deviation of training dataset). The optimized query has a slightly higher LOOD than differing point, and visually has darker color. Our setup uses 500 training data points of the CIFAR10 dataset (50 per class) and NNGP specified by CNN with depth 5.

model). In terms of results, for simple models such as linear model and shallow CNN, they show that the influence function estimates correlates well with leave-one-out retraining. However, for deep neural networks, the influence function worsens significantly ([15, Figure 2] and Figure 3c). By contrast, we show in Figure 3c that mean distance LOOD under NNGP shows high correlation to leave-one-out retraining even for deep neural networks.

Connections between mean distance LOOD and memorization. Memorization, as defined in [6, page 5] and its follow-up variants [7, Equation 1], is given by $\text{mem}(A, S, i) = \Pr_{h \sim A(S)}(h(x_i) = y_i) - \Pr_{h \sim A(S \setminus i)}(h(x_i) = y_i)$; this measures the change of prediction probability on the i th training data after removing it from the training dataset S , given training algorithm A . This again is exactly mean distance LOOD Definition C.1 when query equals the differing data (i.e., self-influence), except that there is an additional mapping (e.g., softmax) from the prediction vector to a binary value that indicates correctness of the prediction. Estimating this memorization precisely requires many retraining runs of the models on the two datasets S and $S \setminus i$ for each pair of training data and dataset. This can lead to a high computational cost; to mitigate this one approach is to resort to empirical influence estimates [7] that use cross-validation-like techniques to reduce the number of retrained models (at a cost of performance drop). By contrast, our mean distance LOOD is efficient as it does not require retraining any NN models (see Footnote 4 for a more detailed computation cost comparison).

G Optimal query under mean distance and deferred proofs for Section 4.3

We discuss here the computation of the optimal query under mean distance. This is equivalent to the question of where the function is most influenced by a change in the dataset.

G.1 Differing point is in general not optimal for Mean Distance LOOD

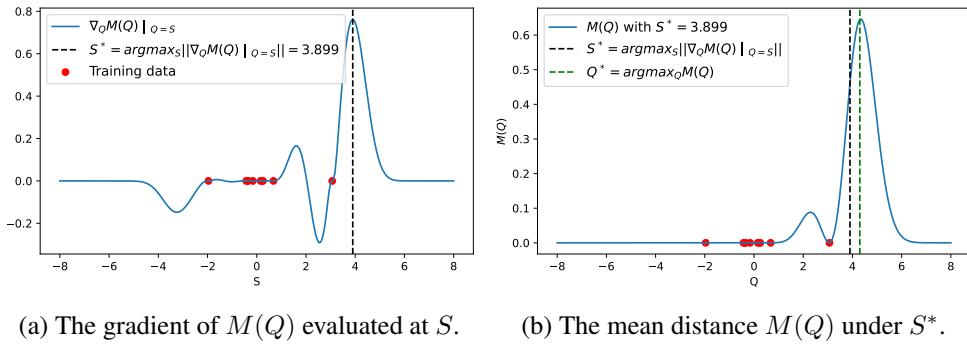
Numerically identifying cases where the differing point is not optimal. We define an algorithm that finds non-optimal differing points S such that when optimizing the query Q , the resulting query is different from S . When the differing point S is not the optimal query, the gradient of $M(Q)$ evaluated at S is non-zero. Hence, intuitively, finding points S such that $\|\nabla_Q M(Q)|_{Q=S}\|$ is maximal should yield

points S for which the optimal query is not the differing point (as there is room for optimising the query due to its positive gradient). We describe this algorithm in Algorithm 1.

Algorithm 1 Identifying non-optimal differing points S

Given D and σ , we solve the optimization problem $S^* = \operatorname{argmax}_S \|\nabla_Q M(Q) |_{Q=S}\|$ to identify a non-optimal differing point. (We use gradient descent to solve this problem.)

A toy example of where the differing point is not optimal. To assess the efficacy of Algorithm 1, we consider a simple example in the one-dimensional case. The setup is as follows: **(1)** Data $D = \{(x_i, y_i), i = 1, \dots, 10\}$ is generated using the following distribution: $x_i \sim \mathcal{N}(0, 1)$, $y_i = \sin(x_i)$. **(2)** Noise parameter is given by $\sigma^2 = 0.01$. **(3)** Kernel is given by $K(x, x') = \exp(-(x - x')^2/2)$ (RBF kernel). **(4)** In Algorithm 1, only S_x is optimized, the label S_y is fixed to $S_y = \sin(S_x)$.



(a) The gradient of $M(Q)$ evaluated at S . (b) The mean distance $M(Q)$ under S^* .

Figure 11: We plot the stationary condition indicator function $\nabla_Q M(Q) |_{Q=S}$ of LOOD when query Q equals the differing data S , over different values of differing data S with $S_x \in [-5, 5]$. This allows us to run Algorithm 1 to identify a differing point $S^* = \max_S \|\nabla_Q M(Q) |_{Q=S}\|$ that has the highest objective (maximal non-stationarity), as shown in Algorithm 1 (where $S^* = 3.899$). In Figure 11b, we validate that under the optimized differing data S^* , the query Q^* that maximizes LOOD (over leave-one-out datasets that differ in S^*) is indeed not the differing point S^* .

We show in Figure 11a the gradient of $M(Q)$ evaluated at S for varying S , and highlight the point S^* (the result of Algorithm 1) that has the largest non-stationarity for mean distance LOOD objective. Additionally, if we look at the query with maximal LOOD under differing data $S = S^*$, the optimal query Q^* that we found is indeed different from the differing point S^* , as shown in Figure 11b.

Let us now look at what happens when we choose other differing points $S \neq S^*$. Figure 12 shows the

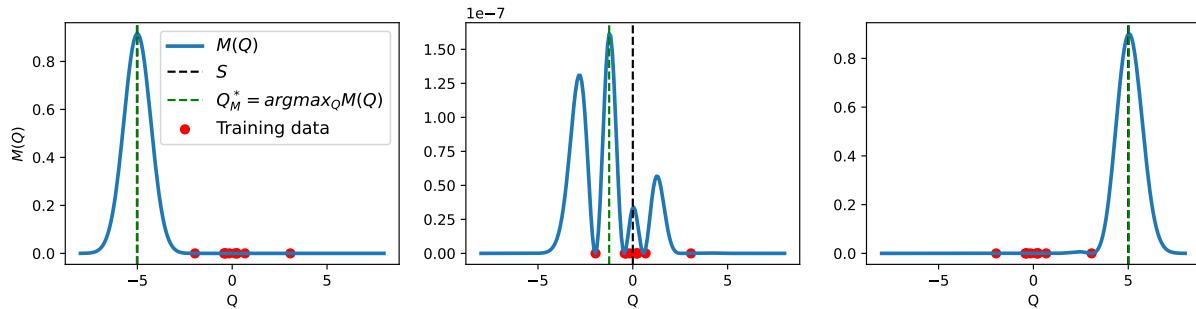


Figure 12: The plot of $M(Q)$ for varying Q and S . We also show the optimal query Q^* and the differing point S .

optimal query for three different choices of S . When S is far from the training data D ($S_x \in \{-5, 5\}$), the optimal query approximately coincides with the differing point S . When S is close to the training

data ($S_x = 0$), the behaviour of the optimal query is non-trivial, but it should be noted that the value of the MSE is very small in this case.

Explaining when the differing point is suboptimal for Mean Distance LOOD. We now give a theoretical explanation regarding why the differing point is suboptimal for mean distance LOOD when it is close (but not so close) to the training dataset (as observed in Figure 12 middle plot). We first give a proposition that shows that the gradient of $M(Q)$ is large when the differing point S is close (but not so close) to the training dataset D .

Proposition 4.1. Consider a dataset D , a differing point S , and a kernel function K satisfying the same conditions of Theorem 3.1. Then, we have that

$$\nabla_Q M(Q) |_{Q=S} = -\alpha^{-2}(1 - K_{SD}M_D^{-1}K_{DS})(y_S - K_{SD}M_D^{-1}y_D)^2 \dot{K}_{SD}M_D^{-1}K_{DS}, \quad (23)$$

where $M(Q)$ denotes the mean distance LOOD as defined in Definition C.1, $K_{SD} \in \mathbb{R}^{1 \times |D|}$ is the kernel matrix between data point S and dataset D , $K_{DS} = K_{SD}^\top$, $\dot{K}_{SD} = \frac{\partial}{\partial Q} K_{QD} |_{Q=S}$, $M_D = K_{DD} + \sigma^2 I$, and $\alpha = 1 - K_{SD}M_D^{-1}K_{DS} + \sigma^2$.

Proof. The left-hand-side of Equation (23) is $\nabla_Q M(Q) = \frac{\partial}{\partial Q} \|\mu(Q) - \mu'(Q)\|_2^2$ by definition of the mean distance LOOD $M(Q)$ in Definition C.1. This term occurs in the term $F_2(Q)$ in Equation (13), in the proof for Theorem 3.1. Specifically, we have $\nabla_Q M(Q) = F_2(S) \cdot \Sigma'(S)$. The detailed expression for $F_2(S)$ is computed in line Equation (21) during the proof for Theorem 3.1. Therefore, by multiplying $F_2(S)$ in Equation (21) with $\Sigma'(S)$, we obtain the expression for $\nabla_Q M(Q)$ in the statement Equation (23). \square

As a result, it should be generally expected that $\nabla_Q M(Q) |_{Q=S} \neq 0$. We can further see that when S is far from the data in the sense that $K_{SD} \approx 0$, the gradient is also close to 0. In the opposite case where S is close to D , e.g. satisfying $y_S \approx K_{SD}M_D^{-1}y_D$, then the gradient is also close to zero. Between the two cases, there exists a region where the gradient norm is maximal.

G.2 Differing point is optimal in Mean Distance LOOD when it is far from the dataset

Explaining when the differing point is optimal for Mean Distance LOOD. We now prove that the differing point is (nearly) optimal when it is far from remaining dataset. We assume the inputs belong to some compact set $C \in \mathbb{R}^d$.

Definition G.1 (Isotropic kernels). We say that a kernel function $K : C \times C \rightarrow \mathbb{R}$ is isotropic if there exists a function $g : \mathbb{R} \rightarrow \mathbb{R}$ such that $K(x, x') = g(\|x - x'\|)$ for all $x, x' \in C$.

Examples includes RBF kernel, Laplace kernel, etc. For this type of kernel, we have the following result.

Lemma G.2. Let K be an isotropic kernel, and assume that its corresponding function g is differentiable on C (or just the interior of C). Then, for all $Q \in C$, we have that

$$\|\nabla_Q M(Q)\| \leq \zeta (d'(Q, S) + d'(Q, D)) \max(d(Q, S), d(Q, D)),$$

where

- $D = \{D_1, \dots, D_n\}$, $D' = D \cup \{S\}$

- $d'(Q, D) = \max_{i \in [n]} |g'(\|Q - D_i\|)|$ and $d'(Q, S) = g'(\|Q - S\|)$
- $d(Q, D) = \max_{i \in [n]} |g(\|Q - D_i\|)|$ and $d(Q, S) = g(\|Q - S\|)$
- $\zeta > 0$ is a constant that depends only on D .

Proof. Let $Q \in C$. For the sake of simplification, we use the following notation: $u = (K_{DD} + \sigma^2 I)^{-1}y_D$, and $u' = (K_{D'D'} + \sigma^2 I)^{-1}y_{D'}$.

Simple calculations yield

$$\nabla_Q M(Q) = (K_{QD'} u' - K_{QDu}) \left(\sum_{i=1}^n (u'_i - u_i) \nabla_Q K_{QD_i} + u'_{n+1} \nabla_Q K_{Q,S} \right).$$

The first term is upperbounded by $\|u'_{[n]} - u\| d(Q, D) + |u'_{n+1}| d(Q, S)$, where $u'_{[n]}$ is the vector containing the first n coordinates of u' , and u_i refers to the i^{th} coordinate of the vector u .

Now observe that $\|u'\|$ is bounded for all S , provided the data labels y are bounded. For this, it suffices to see that $\|(K_{D'D'} + \sigma^2 I)^{-1}\|_2 \leq \sigma^{-2}$ because of the positive semi-definiteness of $K_{D'D'}$.

Moreover, we have $\nabla_Q K_{QD_i} = g'(\|Q - D_i\|) \frac{Q - D_i}{\|Q - D_i\|}$, which implies that $\|\nabla_Q K_{QD_i}\| = |g'(\|Q - D_i\|)|$. Therefore, using the triangular inequality, we obtain

$$\left\| \sum_{i=1}^n (u'_i - u_i) \nabla_Q K_{QD_i} + u'_{n+1} \nabla_Q K_{Q,S} \right\| \leq \|u'_{[n]} - u\|_1 d'(Q, D) + |u'_{n+1}| |g'(\|Q - S\|)|.$$

The existence of ζ is trivial. □

From Lemma G.2, we have the following corollary.

Corollary G.3. Assume that $\sup_{z \in \mathbb{R}} g(z) < \infty$, $\lim_{|z| \rightarrow \infty} g'(z) = 0$ and $g'(0) = 0$. Then, provided that C is large enough, for any $D = \{D_1, \dots, D_n\}$ and $\epsilon > 0$, there exists a point S , such that if we define $D' = D \cup \{S\}$, then $\|\nabla_Q M(S)\| \leq \epsilon$.

That is, if we choose S far enough from the data D , then the point S is close to a stationary point.

Although this shows that gradient is small at S (for S far enough from the data D), it does not say whether this is close to a local minimum or maximum. To determine the nature of the stationary points in this region, an analysis of the second-order geometry is needed. More precisely, we would like to understand the behaviour of the Hessian matrix in a neighbourhood of S . In the next result, we show that the Hessian matrix is strictly negative around S (for S sufficiently far from the data D), which confirms that any stationary point in the neighbourhood of S should be a local maximum.

Lemma G.4 (Second-order analysis). Assume that $g(0) > 0$, $\lim_{z \rightarrow 0} z^{-1} g'(z) = l_-$, where l_- is a negative constant that depends on the kernel function, and that $\lim_{z \rightarrow \infty} z^{-1} g'(z) = 0$. Then, provided that C is large enough, for any $D = \{D_1, \dots, D_n\}$ and $\epsilon > 0$, there exists a point S , such that if we define $D' = D \cup \{S\}$, then there exists $\eta > 0$ satisfying: for all $Q \in \mathcal{B}(S, \eta)$, $\nabla_Q^2 M(Q) \prec 0$.

Lemma G.4. Assume that $g(0) > 0$, $\lim_{z \rightarrow 0} z^{-1} g'(z) = l_-$, where $l_- < 0$ is a negative constant that depends on the kernel function, and that $\lim_{z \rightarrow \infty} z^{-1} g'(z) = 0$. Then, provided that C is large enough, for any $D = \{D_1, \dots, D_n\}$ and $\epsilon > 0$, there exists a point S , such that if we define $D' = D \cup \{S\}$, then there exists $\eta > 0$ satisfying: for all $Q \in \mathcal{B}(S, \eta)$, $\nabla_Q^2 M(Q) \prec 0$ is a negatively definite matrix.

Proof. Let $Q \in C$. Let $v(Q) = K_{QD'}u' - K_{QT}u$. The Hessian is given by

$$\nabla_Q^2 M(Q) = v(Q)\nabla_Q^2 v(Q) + \nabla_Q v(Q)\nabla_Q v(Q)^\top$$

where $\nabla_Q^2 v(Q) = \sum_{i=1}^n (u'_i - u_i) \nabla_Q^2 K_{QD_i} + u'_{n+1} \nabla_Q^2 K_{Q,S}$ and $\nabla_Q v(Q) = \sum_{i=1}^n (u'_i - u_i) \nabla_Q K_{QD_i} + u'_{n+1} \nabla_Q K_{Q,S}$. By Lemma G.2, we have that for S far enough from the dataset D , we have that $\|\nabla_Q v(S)\| \leq \epsilon$. Therefore, the Hessian at S simplifies to the following.

$$\nabla_Q^2|_{Q=S} M(Q) \leq v(Q) \left(\sum_{i=1}^n (u'_i - u_i) \nabla_Q^2 K_{QD_i} + u'_{n+1} \nabla_Q^2 K_{Q,S} \right) + \epsilon^2.$$

For $i \in [n]$, simple calculations yield

$$\nabla_Q^2 K_{QD_i} = \alpha(Q, D_i) \frac{(Q - D_i)(Q - D_i)^\top}{\|Q - D_i\|^2} + \frac{g'(\|Q - D_i\|)}{\|Q - D_i\|} I,$$

where $\alpha(Q, D_i) = g''(\|Q - D_i\|) - \frac{g'(\|Q - D_i\|)}{\|Q - D_i\|}$.

Notice that $u'_{n+1} \approx y_S - K_{SD}u$. Without loss of generality, assume that $y_S > 0$. For S far enough from the dataset D , we then have $u'_{n+1} > 0$ and $v(S) \approx y_S > 0$. Let us now see what happens to the Hessian. For such S , and for any $\epsilon > 0$, there exists a neighbourhood $\mathcal{B}(S, \eta)$, $\eta > 0$, such that for all $Q \in \mathcal{B}(S, \eta)$, we have $\frac{|g'(\|Q - D_i\|)|}{\|Q - D_i\|} < \epsilon$, and $|\alpha(Q, D_i)| < \epsilon$. This ensures that $\|\nabla_Q^2 K_{QD_i}\|_F \leq 2\epsilon$ where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. On the other hand, we have $\nabla_Q^2 K_{Q,S} = \lim_{z \rightarrow 0} \frac{g'(z)}{z} I = l_- I$. It remains to deal with $v(Q)$. Observe that for S far enough from the data D , and Q close enough to S , $v(Q)$ is close to $v(S) \approx y_S > 0$. Thus, taking ϵ small enough, there exists η satisfying the requirements, which concludes the proof. \square

Lemma G.4 shows that whenever the point S is isolated from the dataset D , the Hessian is negative in a neighbourhood of S , suggesting that any local stationary point can only be a local maximum. This is true for isotropic kernels such as the Radial Basis Function kernel (RBF) and the Laplace kernel. As future work, Lemma G.2 and Lemma G.4 have important computational implications for the search for the ‘worst’ poisonous point S . Finding the worst poisonous point S is computationally expensive for two reasons: we cannot use gradient descent to optimize over S since the target function is implicit in S , and for each point S , a matrix inversion is required to find $\max_Q M(Q)$. Our analysis suggests that if S is far enough from the data D , we can use $M(S)$ as a lower-bound estimate of $\max_Q M(Q)$.

G.3 The optimal query for Mean Distance LOOD on image datasets

As we concluded in the analysis in Section 3, only when the differing datapoint is far away (in some notion of distance) from the rest of the dataset is the optimal query (in mean distance LOOD) the differing datapoint. In Figure 11b, we successfully identified settings on a toy dataset where the differing point is not optimal for mean distance LOOD. We now investigate whether this also happens on an image dataset.

In Figure 13, we observe that the optimized record is indeed similar (but not identical) to the differing record for a majority fraction of cases. Therefore, also in image data the removal of a certain datapoint may influence the learned function most in a location that is not the removed datapoint itself. However, the optimized query for Mean distance LOOD still has much smaller MIA AUC score than the differing data. This again indicates that MSE optimization finds a suboptimal query for information leakage, despite achieving higher influence (mean distance).

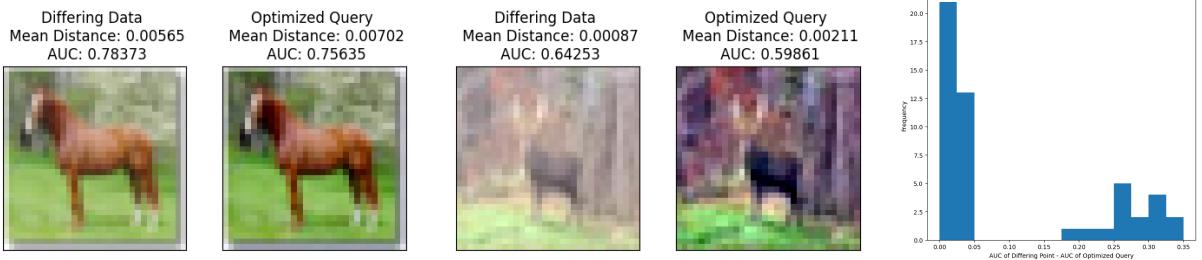


Figure 13: Results for running single query optimization for Mean distance LOOD on class-balanced subset of CIFAR-10 dataset of size 1000 and RBF kernel. The optimized query has higher mean distance LOOD than the differing data record, despite having lower MIA AUC score. Across the 50 trials that we evaluated, more than 50% of trials result in an optimized query that has lower AUC score (by more than 0.025) than the differing data.

H Deferred proofs for Section 5

Our starting point is the following observation in prior work [11], which states that the NNGP kernel is sensitive to the choice of the activation function.

Proposition H.1. *Assume that the inputs are normalized, $\|x\| = 1$ and let K_{NNGP}^L denote the NNGP kernel function of a an MLP of depth $L \geq 1$. Then, the for any two inputs, we have the following*

- [Non-smooth activations] With ReLU: there exists a constant $\alpha > 0$ such that $|k_{NNGP}^L(x, x') - \alpha| = \Theta(L^{-2})$.
- [Smooth activations] With GeLU, ELU, Tanh: there exists a constant $\alpha > 0$ such that $|k_{NNGP}^L(x, x') - \alpha| = \Theta(L^{-1})$.

As a result, for the same depth L , one should expect that NNGP kernel matrix with ReLU activation is closer to a low rank constant matrix than with smooth activations such as GeLU, ELU, Tanh.⁸

For proving Proposition 5.1, we will need the following lemma.

Lemma H.2. *Let A and B be two positive definite symmetric matrices. Then, the following holds*

$$\|A^{-1} - B^{-1}\|_2 \leq \max(\lambda_{\min}(A)^{-1}, \lambda_{\min}(B)^{-1})\|A - B\|_2,$$

where $\lambda_{\min}(A)$ is the smallest eigenvalue of A , and $\|\cdot\|_2$ is the usual 2-norm for matrices.

Proof. Define the function $f(t) = (tA + (1-t)B)^{-1}$ for $t \in [0, 1]$. Standard results from linear algebra yield

$$f'(t) = -f(t)(A - B)f(t)$$

⁸The different activations are given by GeLU(x) = $x\Phi(x)$ where Φ is the c.d.f of a standard Gaussian variable, ELU(x) = $1_{\{x \geq 1\}}x + 1_{\{x < 1\}}(e^x - 1)$, and Tanh(x) = $(e^x - e^{-x})(e^x + e^{-x})^{-1}$.

Therefore, we have

$$\begin{aligned}
\|f(1) - f(0)\|_2 &= \left\| \int_0^1 f'(t) dt \right\|_2 \\
&\leq \int_0^1 \|f'(t)\|_2 dt \\
&\leq \int_0^1 \|f(t)\|_2^2 \|A - B\|_2 dt \\
&= \int_0^1 (\lambda_{\min}(tA + (1-t)B))^{-2} \|A - B\|_2 dt.
\end{aligned}$$

Using Weyl's inequality on the eigenvalues of the sum of two matrices, we have the following

$$\begin{aligned}
\lambda_{\min}(tA + (1-t)B) &\geq \lambda_{\min}(tA) + \lambda_{\min}((1-t)B) \\
&= t\lambda_{\min}(A) + (1-t)\lambda_{\min}(B) \\
&\geq \min(\lambda_{\min}(A), \lambda_{\min}(B)).
\end{aligned}$$

Combining the two inequalities concludes the proof. \square

We are now ready to prove Proposition 5.1, as follows.

Proposition 5.1. Let D and $D' = D \cup S$ be an arbitrary pair of leave-one-out datasets, where D contains n records and D' contains $n+1$ records. Define the function $h(\alpha) = \sup_{x,x' \in D'} |K(x, x') - \alpha|$. Let $\alpha_{\min} = \operatorname{argmin}_{\alpha \in \mathbb{R}^+} h(\alpha)$, and thus $h(\alpha_{\min})$ quantifies how close the kernel matrix is to a low rank all-constant matrix. Assume that $\alpha_{\min} > 0$. Then, it holds that

$$\max_Q M(Q) \leq C_{n,\alpha_{\min},\zeta} n^{5/2} h(\alpha_{\min}) + C_{\alpha,\zeta} n^{-1},$$

where

$$C_{n,\alpha_{\min},\zeta} = \max \left(\frac{(\alpha_{\min} + h(\alpha_{\min}))\zeta}{\min(\alpha_{\min}\sigma^2, |\alpha_{\min}\sigma^2 - nh(\alpha_{\min})|)}, \frac{(1+n^{-1})^{5/2}(\alpha_{\min} + h(\alpha_{\min}))\zeta}{\min(\alpha_{\min}\sigma^2, |\alpha_{\min}\sigma^2 - (n+1)h(\alpha_{\min})|)} \right),$$

and $C_{\alpha_{\min},\zeta} = \zeta(1 + \alpha_{\min}^{-1})$.

As a result, if $n^{5/2}h(\alpha_{\min}) \ll 1$ and $n \gg 1$, then the maximum mean distance LOOD (i.e., maximum influence) satisfies $\max_Q M(Q) \ll 1$.

Similarly, there exist constants $A_n, B > 0$ such that

$$\max_Q KL(f|_{D,\sigma^2}(Q) \| f|_{D',\sigma^2}(Q)) \leq A_n h(\alpha_{\min}) + B n^{-1}.$$

Proof. Let Q be a query point and $\alpha \in \mathbb{R}$. We have the following

$$M(Q) = \frac{1}{2} \| K_{QD} M_D^{-1} y_D - K_{QD'} M_{D'}^{-1} y_{D'} \|,$$

where $M_D = (K_{DD} + \sigma^2 I)$ and the same holds for $M_{D'}$. Let $U = \alpha U_n + \sigma^2 I$ and $U' = \alpha U_{n+1} + \sigma^2 I$, where U_n is the $n \times n$ matrix having ones everywhere. With this notation, we have the following bound

$$2 \times M(Q) \leq \|K_{QD}(M_D^{-1} - U^{-1})y_D\| + \|K_{QD'}(M_{D'}^{-1} - (U')^{-1})y_{D'}\| \\ + \|K_{QD'}(U')^{-1}y_{D'} - K_{QD}U^{-1}y_D\|,$$

Let us deal with first term. We have that

$$\|K_{QD}(M_D^{-1} - U^{-1})y_D\| \leq \|K_{QD}\| \|y_D\| \|M_D^{-1} - U^{-1}\| \leq n^{1/2}(\alpha + h(\alpha)) \|y_D\|_2 \|M_D^{-1} - U^{-1}\|.$$

Using Lemma H.2, we have the following

$$\|M_D^{-1} - U^{-1}\| \leq \max(\lambda_{\min}(A)^{-1}, \lambda_{\min}(B)^{-1}) \|M_D - U\|,$$

which implies that

$$\|K_{QD}(M_D^{-1} - U^{-1})y_D\| \leq \frac{n^{3/2}(\alpha + h(\alpha)) \|y_D\|_2}{\min(\alpha\sigma^2, |\alpha\sigma^2 - nh(\alpha)|)} \times h(\alpha),$$

where we have used the inequality $|\lambda_{\min}(A) - \lambda_{\min}(B)| \leq \|A - B\|_2$ for any two symmetric positive matrices A, B .

Similarly, we have the following

$$\|K_{QD'}(M_{D'}^{-1} - (U')^{-1})y_{D'}\| \leq \frac{(n+1)^{3/2}(\alpha + h(\alpha)) \|y_{D'}\|_2}{\min(\alpha\sigma^2, |\alpha\sigma^2 - (n+1)h(\alpha)|)} \times h(\alpha).s$$

For the last term, we have

$$\begin{aligned} \|K_{QD'}(U')^{-1}y_{D'} - K_{QD}U^{-1}y_D\| &\leq \|(K_{QD'} - \alpha e_{n+1}^\top)(U')^{-1}y_{D'}\| \\ &\quad + \|(K_{QD} - \alpha e_n^\top)U^{-1}y_D\| \\ &\quad + \|\alpha e_{n+1}^\top(U')^{-1}y_{D'} - \alpha e_n^\top U^{-1}y_D\|. \end{aligned}$$

Moreover, we have

$$\|(K_{QD} - \alpha e_n^\top)U^{-1}y_D\| \leq \sigma^{-2} \|y_D\| \sqrt{n} h(\alpha),$$

and,

$$\|(K_{QD'} - \alpha e_{n+1}^\top)(U')^{-1}y_{D'}\| \leq \sigma^{-2} \|y_{D'}\| \sqrt{n+1} h(\alpha).$$

For the last term, we have

$$\begin{aligned} \|\alpha e_{n+1}^\top(U')^{-1}y_{D'} - \alpha e_n^\top U^{-1}y_D\| &\leq \frac{\alpha}{|\alpha n + \sigma^2| |\alpha(n+1) + \sigma^2|} \left| \sum_{i=1}^n y_{D_i} \right| + \frac{\alpha}{|\alpha(n+1) + \sigma^2|} |y_S| \\ &\leq (1 + \alpha^{-1}) \zeta n^{-1}. \end{aligned}$$

Combining all these inequalities yield the desired bound.

For the second upperbound, observe that $KL(f|_{D,\sigma^2}(Q)\|f|_{D',\sigma^2}(Q)) = \log\left(\frac{\Sigma'(Q)}{\Sigma(Q)}\right) + \frac{\Sigma(Q)}{2\Sigma'(Q)} + \frac{(\mu(Q)-\mu(Q'))^2}{\Sigma'(Q)} - \frac{1}{2}$ by the definition of KL LOOD in Definition C.3. Meanwhile, for sufficiently small $h(\alpha_{min})$, we have that

$$\Sigma(Q) = \alpha_{min} - \alpha_{min}^2 n (\sigma^2 + n\alpha_{min})^{-1} + \mathcal{O}(h(\alpha_{min})) = \alpha_{min} \sigma^2 (\sigma^2 + n\alpha_{min})^{-1} + \mathcal{O}(h(\alpha_{min})),$$

and a similar formula holds for $\Sigma'(Q)$ with $(n+1)$ instead of n . Therefore, we obtain

$$\frac{\Sigma(Q)}{\Sigma'(Q)} = 1 + \frac{\alpha_{min}}{\sigma^2 + \alpha_{min} n}.$$

Combining this result with the upperbound on the mean distance, the existence of A_n and B is straightforward. \square

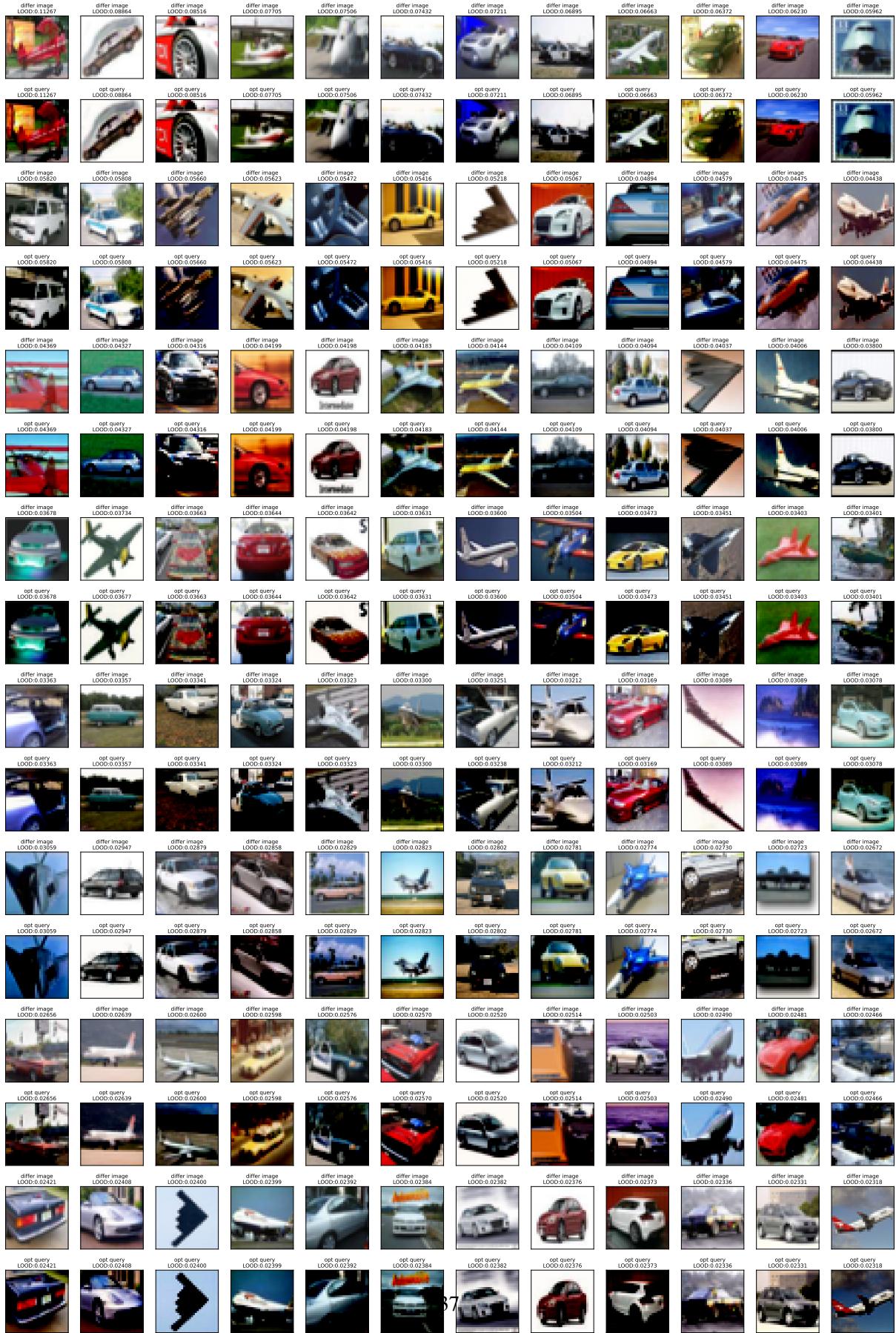


Figure 14: Visualization of 500 **randomly chosen** differing data and query optimized by LOOD (Part 1). We show differing data above the optimized query, and use their LOOD gap to sort the images.

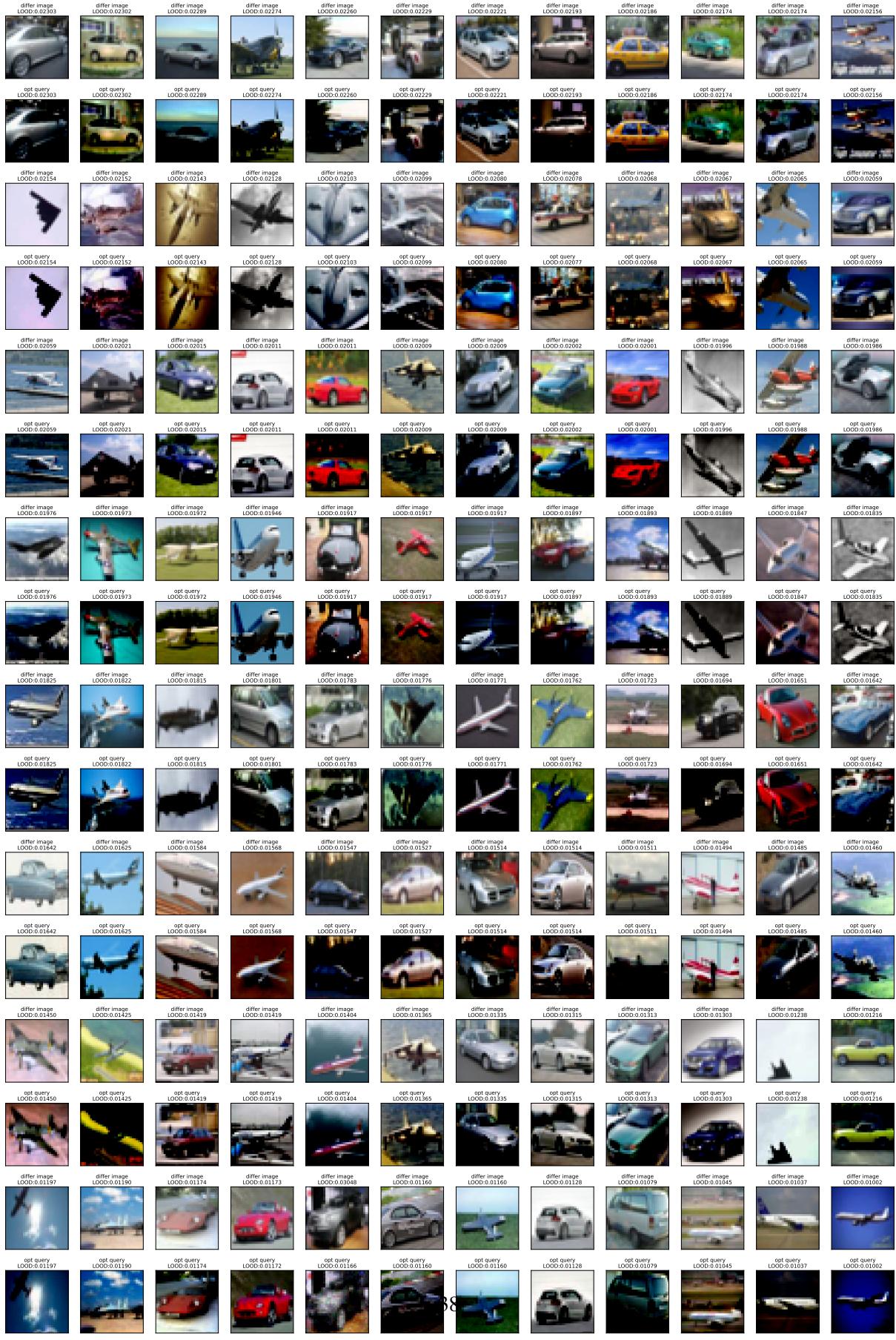


Figure 15: Visualization of 500 **randomly chosen** differing data and query optimized by LOOD (Part 2). We show differing data above the optimized query, and use their LOOD gap to sort the images.

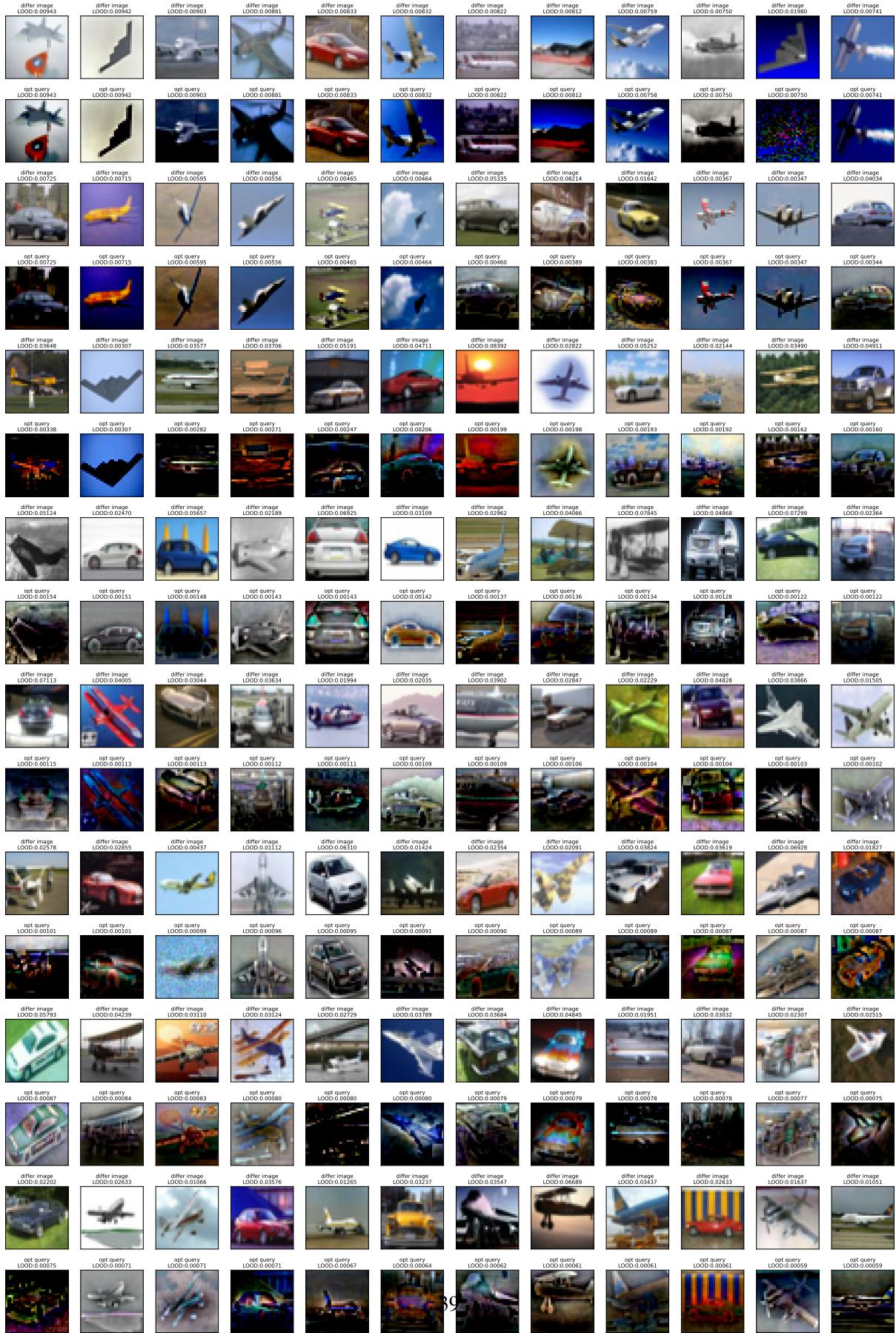


Figure 16: Visualization of 500 **randomly chosen** differing data and query optimized by LOOD (Part 3). We show differing data above the optimized query, and use their LOOD gap to sort the images.

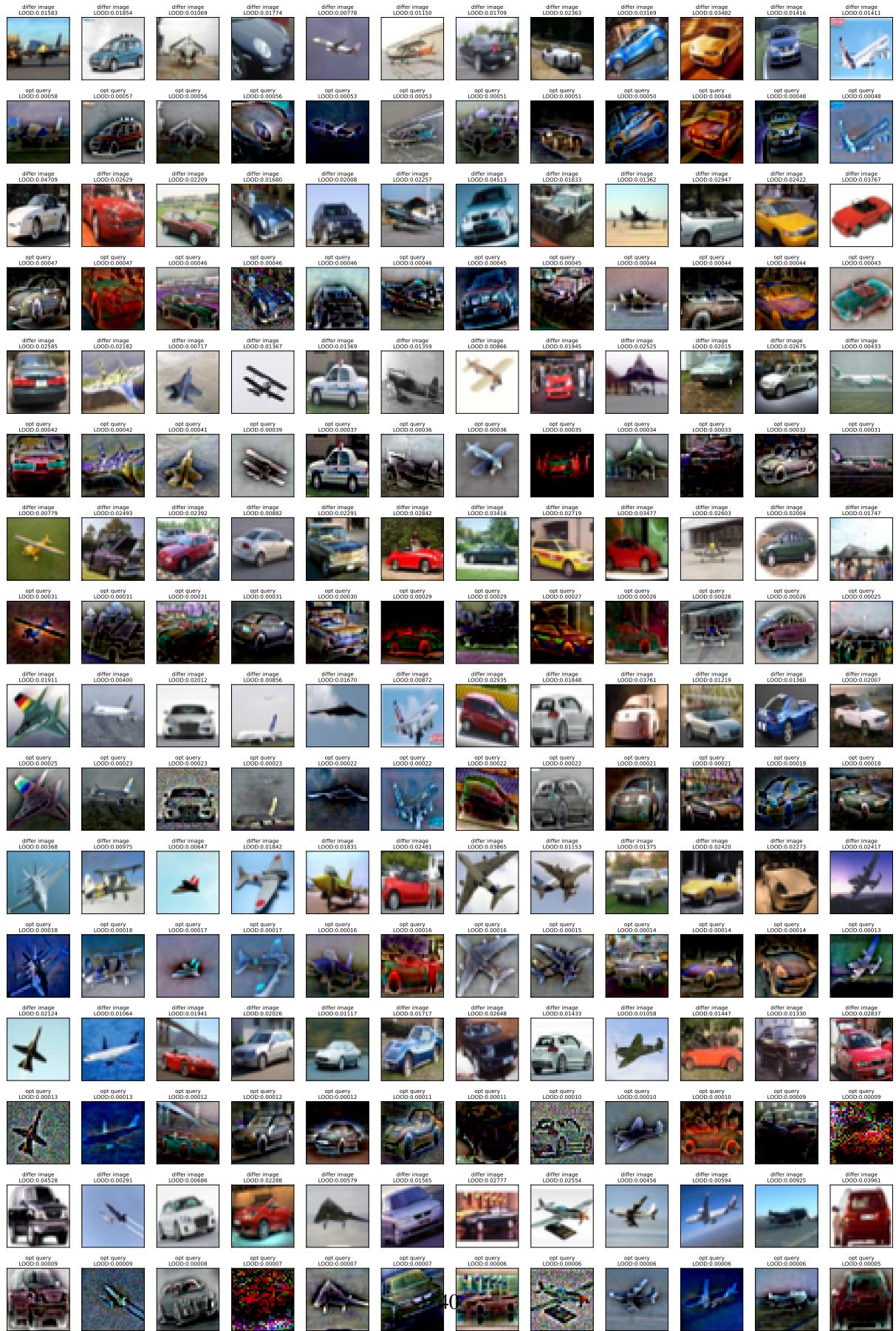


Figure 17: Visualization of 500 **randomly chosen** differing data and query optimized by LOOD (Part 4). We show differing data above the optimized query, and use their LOOD gap to sort the images.

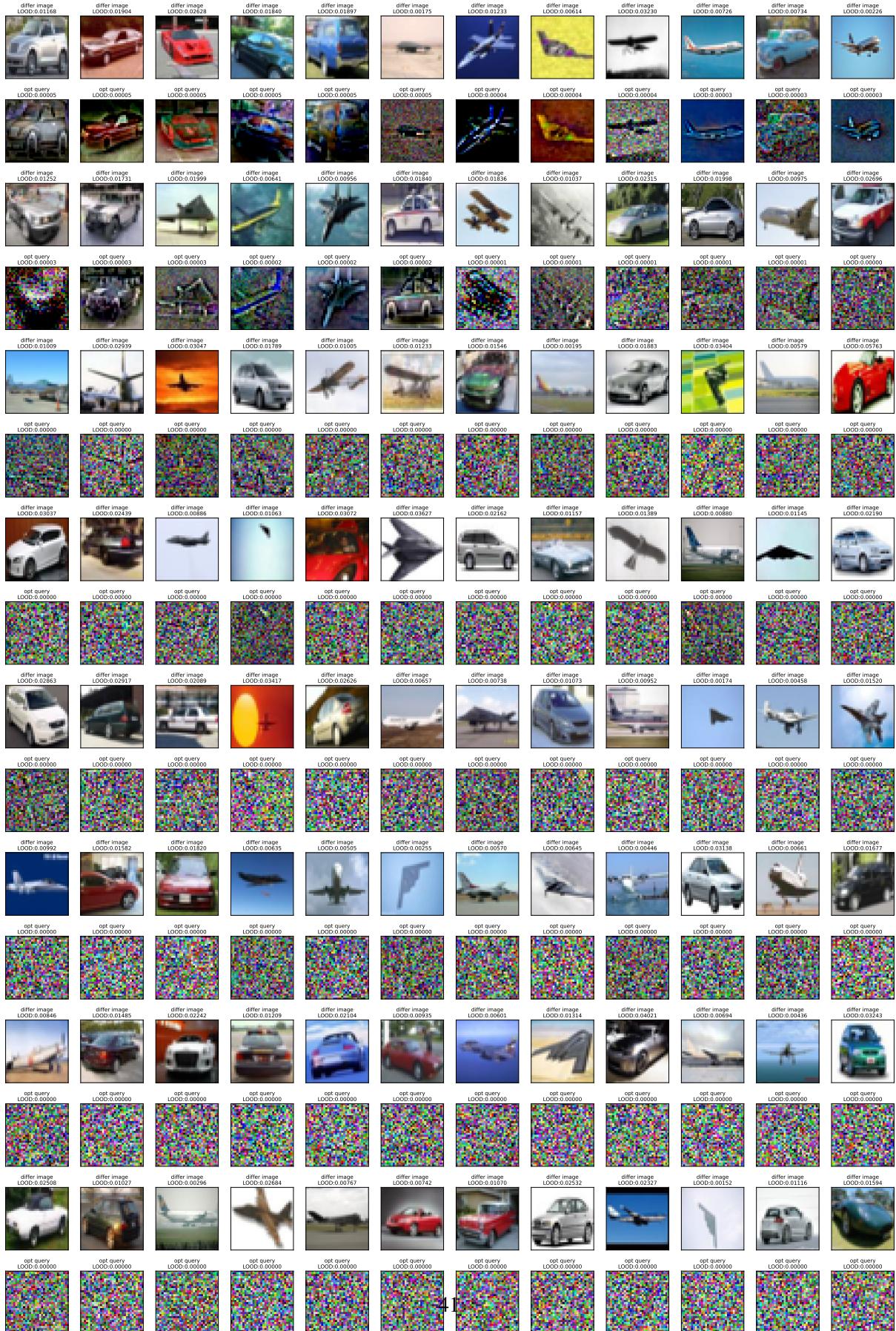


Figure 18: Visualization of 500 **randomly chosen** differing data and query optimized by LOOD (Part 5). We show differing data above the optimized query, and use their LOOD gap to sort the images.

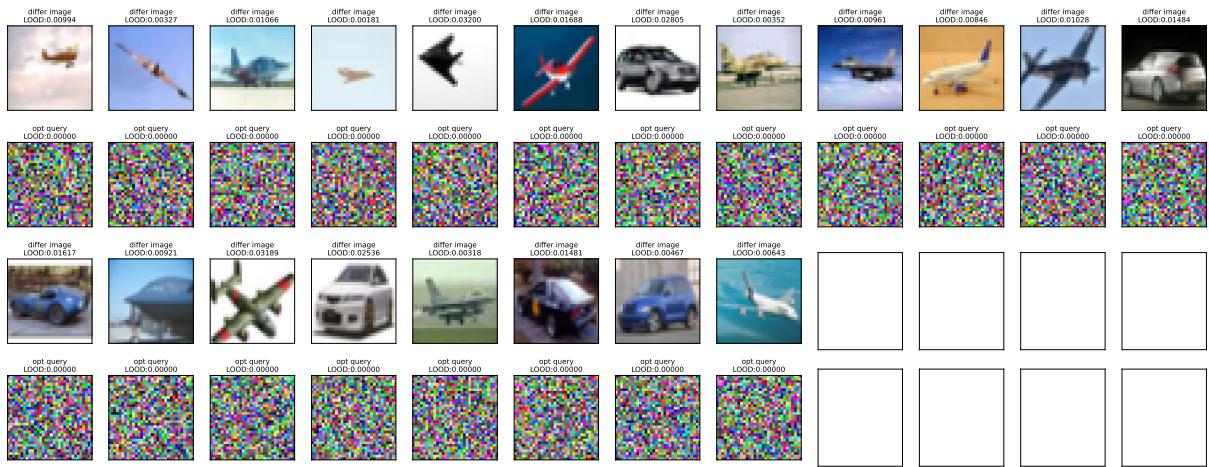


Figure 19: Visualization of 500 **randomly chosen** differing data and query optimized by LOOD (Part 6). We show differing data above the optimized query, and use their LOOD gap to sort the images.