Unveiling Privacy, Memorization, and Input Curvature Links

Deepak Ravikumar [1] **Efstathia Soufleri** [1] **Abolfazl Hashemi** [1] **Kaushik Roy** [1]

Abstract

Deep Neural Nets (DNNs) have become a pervasive tool for solving many emerging problems. However, they tend to overfit to and memorize the training set. Memorization is of keen interest since it is closely related to several concepts such as generalization, noisy learning, and privacy. To study memorization, Feldman (2019) proposed a formal score, however its computational requirements limit its practical use. Recent research has shown empirical evidence linking input loss curvature (measured by the trace of the loss Hessian w.r.t inputs) and memorization. It was shown to be ~ 3 orders of magnitude more efficient than calculating the memorization score. However, there is a lack of theoretical understanding linking memorization with input loss curvature. In this paper, we not only investigate this connection but also extend our analysis to establish theoretical links between differential privacy, memorization, and input loss curvature. First, we derive an upper bound on memorization characterized by both differential privacy and input loss curvature. Second, we present a novel insight showing that input loss curvature is upper-bounded by the differential privacy parameter. Our theoretical findings are further validated using deep models on CIFAR and ImageNet datasets, showing a strong correlation between our theoretical predictions and results observed in practice.

1. Introduction

Machine learning and deep learning approaches have become state-of-the-art solutions in many learning tasks such as computer vision, natural language processing, etc. However, Deep Neural Nets (DNNs) are prone to over-fitting and memorization. An increasingly larger number of recent literature has focused on understanding memorization in DNNs (Zhang et al., 2017; Arpit et al., 2017; Carlini et al., 2019; Feldman & Vondrak, 2019; Feldman & Zhang, 2020; Feldman, 2019). This is crucial given the implications to several connected areas such as generalization (Zhang et al., 2021; Brown et al., 2021), noisy learning (Liu et al., 2020), identifying mislabelled examples (Maini et al., 2022), identifying rare and hard examples (Carlini et al., 2019), privacy (Feldman, 2019), risks from membership inference attacks (Shokri et al., 2017; Carlini et al., 2022) and more.

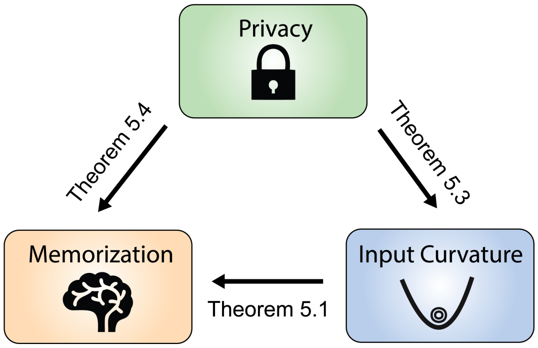


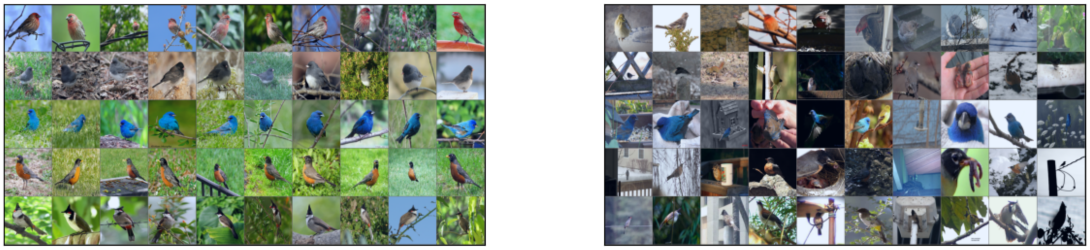
Figure 1. Our theoretical framework provides upper bounds in Theorems 5.1, 5.3, and 5.4. These are visualized as links between Differential Privacy, Memorization, and Input Loss Curvature.

To study memorization several metrics have been suggested. Carlini et al. (2019) proposed a combination of five metrics to analyze memorization. Alternatively, Jiang et al. (2020) proposed using a computationally efficient proxy to C-score, a metric closely related to the stability-based memorization (Feldman, 2019). The stability-based memorization score proposed by Feldman (2019) measures the change in expected output probability when the sample under investigation is removed from the training dataset. Additionally, unlike other proposed metrics, Feldman (2019) provides a strong theoretical framework for understanding memorization. This theory was then tested in practice in a subsequent paper (Feldman & Zhang, 2020). However, their method involved training thousands of models and is thus computationally infeasible in most real applications.

In a recent paper, Garg et al. (2023) suggested a new proxy using input loss curvature to measure the stability-based memorization score proposed in Feldman (2019). To measure input loss curvature they suggested using the trace of

[1] Department of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana 47906. Correspondence to: Deepak Ravikumar <dravikum@purdue.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).



(a) Low curvature examples from ImageNet.

(b) High curvature examples from ImageNet.

Figure 2. Images from ImageNet ranked using input loss curvature. Input loss curvature was obtained using a single ResNet18 trained on ImageNet. Ten lowest curvature samples (left) and ten highest curvature samples (right) from the training set are visualized for 5 classes (each row is a class) from ImageNet. Low curvature samples are 'prototypical' of their class, while high curvature samples are rare, difficult, and more likely memorized instances.

the loss Hessian with respect to the input. Using this input loss curvature measurement, they provided evidence on the link between memorization and input loss curvature. They obtained high cosine similarity between input loss curvature and memorization scores from Feldman & Zhang (2020) while being ~ 3 orders of magnitude less compute-intensive. To illustrate the savings we reproduced Garg et al. (2023)'s results on ImageNet and visualized the ten lowest and highest curvature samples in Figure 2. These examples were obtained using a *single* ResNet18 model trained on ImageNet, compared to 1000's of models trained by Feldman & Zhang (2020) to compute memorization scores. From Figure 2, we see that low curvature samples are 'prototypical' of their class. While high curvature samples are drawn from rare, hard, or outlier examples which are more likely to be memorized.

Input loss curvature is thus, a promising proxy for stability-based memorization score. However, there is a lack of theoretical understanding of the link between memorization, and input loss curvature. In this paper, we develop a theoretical framework to understand this observation and formally unveil the connections between memorization and input loss curvature. Further, we explore the relationship beyond memorization and input loss curvature, our theoretical contributions reveal the underlying link between differential privacy (Dwork et al., 2006), memorization (Feldman, 2019), and input loss curvature (Garg et al., 2023). We present the links as three theorems. The first links memorization and input loss curvature, and the second theorem links input loss curvature and differential privacy. The third theorem links differential privacy and memorization. These links are visualized in Figure 1. Each of the three theoretical links developed in this paper is corroborated by evidence obtained on DNNs used for vision classification tasks (code available at this github link).

In summary, the main contributions of this paper are given below:

- We develop a theoretical framework for analyzing input loss curvature and memorization in a general learning setting and demonstrate its implications to DNNs.
- We show that memorization is upper bounded by (a) input loss curvature and (b) relevant privacy parameters. We also show that input loss curvature is also upper bounded by privacy, completing the theoretical links between memorization, privacy, and input loss curvature.
- We verify the theoretical results with extensive experiments on vision classification tasks using DNNs on CIFAR100 and ImageNet datasets.
- We obtain a tighter bound on private learnability. Namely, we establish that ϵ-differential privacy implies $L(1 - e^{-\epsilon})$ stability, thereby improving the previous theoretical bound.

2. Preliminaries and Notation

Consider a supervised learning problem where the goal is to learn a mapping from some input space $\mathcal{X} \subset \mathbb{R}^d$ to an output space $\mathcal{Y} \subset \mathbb{R}$. The learning is performed using a randomized algorithm $\mathcal{A}$ on a training set S. A randomized algorithm employs a degree of randomness as a part of its logic. The training set S contains m elements. Each element $z_i = (x_i, y_i)$ is drawn from an unknown distribution $\mathcal{D}$, where $z_i \in \mathcal{Z}, x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$ and $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. Thus we define the training set $S \in \mathcal{Z}^m$ as $S = \{z_1, \cdots, z_m\}$. We assume $m \geq 2$. Another very relevant concept is adjacent datasets. Adjacent datasets are obtained when the i^{th} element is removed. This is sometimes referred to as a leave-one-out set defined as

$$S^{\setminus i} = \{z_1, \cdots, z_{i-1}, z_{i+1}, \cdots, z_m\}.$$

Related to adjacent (a.k.a neighboring) datasets is the distance between datasets. The distance between any two datasets S, S' denoted by $\|S - S'\|_1$ is a measure of how many samples differ between S and S'. Note, $\|S\|_1$ denotes the size of a dataset S.

A randomized learning algorithm $\mathcal{A}$ when applied on a dataset S results in a hypothesis denoted by $h_S^\phi = \mathcal{A}(\phi, S)$, where $\phi \sim \Phi$ is the random variable associated with the randomness of the algorithm $\mathcal{A}$. A cost function $c : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}^+$ is used to measure the performance of the hypothesis. The cost of the hypothesis h at a sample z_i is also referred to as the loss ℓ at z_i defined as

$$\ell(h, z_i) = c(h(x_i), y_i).$$

In most cases, we are interested in the loss of h over the data distribution, which is referred to as population risk defined as

$$R(h) = \mathbb{E}_{z\sim\mathcal{D}}[\ell(h, z)].$$

Since the data distribution $\mathcal{D}$ is unknown in general, it is common to evaluate and study the empirical risk defined as

$$R_{emp}(h, S) = \frac{1}{m}\sum_{i=1}^{m} \ell(h, z_i), \quad z_i \in S.$$

In this paper, our characterization of curvature involves the gradient and the Hessian of the loss *with respect to the input data*, which is denoted using the ∇ and ∇^2 operators respectively. $\|a\|$ denotes the ℓ_2 norm of a.

3. Background

Differential Privacy was introduced by Dwork et al. (2006) and here we briefly recall the definition. A randomized algorithm $\mathcal{A}$ with domain $\mathcal{Z}^m$ is ϵ-differentially private if for all $\mathcal{R} \subset \text{Range}(\mathcal{A})$ and for all $S, S' \in \mathcal{Z}^m$ such that $||S - S'||_1 \leq 1$

$$\Pr_\phi[h_S^\phi \in \mathcal{R}] \leq e^\epsilon \Pr_\phi[h_{S'}^\phi \in \mathcal{R}], \tag{1}$$

where the probability is taken over the randomness arising from the algorithm $\mathcal{A}, \phi \sim \Phi$.

Memorization of the i^{th} element $z_i = (x_i, y_i)$ of the dataset S by an algorithm $\mathcal{A}$ was defined by Feldman (2019) using the notion of stability as:

$$\text{mem}(\mathcal{A}, S, i) = \Pr_\phi[h_S^\phi(x_i) = y_i] - \Pr_\phi[h_{S^{\setminus i}}^\phi(x_i) = y_i], \tag{2}$$

where the probability is taken over the randomness of algorithm $\mathcal{A}$.

Error Stability of a possibly randomized algorithm $\mathcal{A}$ for some $\beta > 0$ is defined as Kearns & Ron (1997)

$$\forall i \in \{1, \cdots, m\}, \ \left|\mathbb{E}_{\phi,z}[\ell(h_S^\phi, z)] - \mathbb{E}_{\phi,z}[\ell(h_{S^{\setminus i}}^\phi, z)]\right| \leq \beta, \tag{3}$$

where $z \sim \mathcal{D}$ and $\phi \sim \Phi$.

Generalization. A randomized algorithm $\mathcal{A}$ is said to generalize with confidence δ and a rate $\gamma'(m)$ if

$$\Pr[|R_{emp}(h, S) - R(h)| \leq \gamma'(m)] \geq \delta. \tag{4}$$

Uniform Model Bias. The hypothesis h resulting from the application of algorithm $\mathcal{A}$ to learn the true conditional $h^* = \mathbb{E}[y|x]$ from a dataset $S \sim \mathcal{D}^m$ has uniform bound on model bias given by Δ if

$$\forall S \sim \mathcal{D}^m, \quad \left|\mathbb{E}_\phi[R(h_S^\phi) - R(h^*)]\right| \leq \Delta. \tag{5}$$

ρ-Lipschitz Hessian. The Hessian of ℓ is Lipschitz continuous on $\mathcal{Z}$ if $\forall z_1, z_2 \in \mathcal{Z}$, and $\forall h \in \text{Range}(\mathcal{A})$ if there exists some $\rho > 0$ such that

$$\|\nabla_{z_1}^2 \ell(h, z_1) - \nabla_{z_2}^2 \ell(h, z_2)\| \leq \rho \|z_1 - z_2\|. \tag{6}$$

Input Loss Curvature. Using the notation of curvature from Moosavi-Dezfooli et al. (2019); Garg et al. (2023), input loss curvature is defined as the sum of the eigenvalues of the Hessian H of the loss with respect to input z_i, conveniently it can be written using the trace as

$$\text{Curv}_\phi(z_i, S) = \text{tr}(H) = \text{tr}(\nabla_{z_i}^2 \ell(h_S^\phi, z_i)) \tag{7}$$

υ-adjacency. A dataset S is said to contain υ-adjacent (read as upsilon-adjacent) elements if it contains two elements z_i, z_j such that $z_j = z_i + \alpha$ for some $\alpha \in B_p(\upsilon)$ (read as υ-Ball). Note that this can be ensured through construction. Consider a dataset S' which has no z_j s.t $z_j = z_i + \alpha; z_j, z_i \in S'$. Then we can construct S such that $S = \{z \,|\, z \in S'\} \cup \{z_i + \alpha\}$ for some $z_i \in S', \alpha \in B_p(\upsilon)$, ensuring υ-adjacency holds.

4. Related Work

Input Loss Curvature is a measure of the sensitivity of the model to a specific input. Loss curvature with respect to weight parameters has received significant attention (Keskar et al., 2017; Wu et al., 2020; Jiang* et al., 2020; Foret et al., 2021; Kwon et al., 2021; Andriushchenko & Flammarion, 2022), recently regarding its role in characterizing the sharpness of a learning objective and its connection to generalization. However, input loss curvature has received less focus. Input loss curvature has been studied in the context of adversarial robustness (Fawzi et al., 2018; Moosavi-Dezfooli et al., 2019), coresets (Garg & Roy, 2023) and recently as a proxy for memorization (Garg et al., 2023). Moosavi-Dezfooli et al. (2019) showed that adversarial training decreases the curvature of the loss surface with respect to inputs. Further, they provided a theoretical link between robustness and curvature and proposed using curvature regularization. Garg & Roy (2023) identified samples with low curvature as being more data-efficient and developed a coreset identification and training algorithm based on input loss curvature. In an interesting application of input loss curvature, Garg et al. (2023) provided evidence linking memorization and input loss curvature.

Memorization has garnered increasing research effort with several recent works aiming to add to the understanding of memorization and its implications (Zhang et al., 2017; Arpit et al., 2017; Carlini et al., 2019; Feldman & Vondrak, 2019; Feldman, 2019; Feldman & Zhang, 2020; Maini et al., 2022; Garg et al., 2023; Lukasik et al., 2023). The motivation for studying memorization stems from a variety of goals ranging from deriving insights into generalization (Zhang et al., 2017; Toneva et al., 2019; Brown et al., 2021; Zhang et al., 2021), identifying mislabeled examples (Pleiss et al., 2020; Maini et al., 2022), and identifying challenging or rare sub-populations (Carlini et al., 2019), to understanding privacy (Feldman, 2019) and robustness risks from memorization (Shokri et al., 2017; Carlini et al., 2022). While several metrics have been proposed to study memorization (Carlini et al., 2019; Jiang et al., 2020), the stability-based memorization score proposed by Feldman (2019) provides a framework to understand memorization (Feldman & Zhang, 2020). However, since the score proposed by Feldman (2019) is computationally expensive, Garg et al. (2023) proposed using input loss curvature as a more compute-efficient proxy. In this paper, we develop the theoretical framework to understand the links between input loss curvature, memorization, and differential privacy.

Influence Functions were applied to deep learning by Koh & Liang (2017) and are closely related to memorization. Influence functions aim to identify the impact of one training point on the model predictions. Influence functions try to answer the counterfactual: what would have happened if a training point were absent, or if its values were changed slightly? While recent approaches (Schioppa et al., 2022) have applied influence functions to large deep models, influence functions have been criticized (Basu et al., 2021; Bae et al., 2022; Schioppa et al., 2023) since the underlying theory assumes strong convexity and positive definiteness of the Hessian, conditions that are not met in the context of DNNs. On the other hand, the theoretical framework we present in this paper does not make any assumptions about the convexity or the definiteness of the Hessian and is more suitable for studying deep learning.

5. Linking Privacy, Memorization and Input Loss Curvature

In this section, we discuss our theoretical contributions as three links. First, we present Theorem 5.1 which links memorization and curvature. Second, we present Theorem 5.3 which links privacy and curvature. Finally, we present Theorem 5.4 which links memorization and privacy.

5.1. Memorization and Input Loss Curvature

The association between memorization and input loss curvature may initially appear counterintuitive at first, but a closer examination reveals a fundamental connection. Both metrics intrinsically capture the sensitivity of a model to input perturbations. Here we provide a theoretical link between memorization and input curvature in the form of Theorem 5.1. Theorem 5.1 is one of our core contributions.

Theorem 5.1 (Curvature Upper Bounds Memorization)**.** *Let the assumptions of error stability 3, generalization 4, and uniform model bias 5 hold and assume the υ-adjacency of the dataset and that the loss is bounded such that $0 \leq \ell \leq L$. Then with probability at least $1-\delta$ it holds*

$$|\text{mem}(\mathcal{A}, S, i)| \leq \frac{1}{L}\mathbb{E}_{\phi}[\text{Curv}_{\phi}(z_i, S^{\setminus i})] + c_1 \tag{8}$$

$$c_1 = \frac{\rho}{6L}\mathbb{E}_{\alpha}[\|\alpha\|^3] + \frac{m\beta}{L} + \frac{(4m-1)\gamma}{L} + \frac{2(m-1)\Delta}{L} \tag{9}$$

Sketch of Proof. Using the result from Nesterov & Polyak (2006) we obtain an upper bound on the loss at z_j involving the Hessian of the loss. By choosing α such that $\mathbb{E}[\alpha] = 0$ we get rid of the first-order terms. Then by taking expectation over the randomness of the algorithm and then performing algebraic manipulation we can show that the expected difference in loss at z_i for two different models is upper bound by the result in Theorem 5.1. The final step is to make the connection that for bounded loss, the difference in loss at z_i for two different models is a scaled version of memorization. The full proof is provided in Appendix A.3.

Interpreting the Theory. Theorem 5.1 (Equation 8) indicates a *linear relationship* between memorization and input loss curvature. Observe that the upper bound is dependent on the input loss curvature of a sample z_i and the offset factor c_1. However, the offset factor c_1 is data independent, i.e. c_1 has no dependence on z_i. The offset factor c_1 (Equation 9) consists of the following components. The third moment of the perturbation random variable α, which is a measure of the skewness of the distribution. By choosing the distribution of α carefully, e.g. a centralized Gaussian, this can be made zero. The second and third terms of Equation 9 are properties of the training algorithm, i.e. the algorithm's stability β and ability to generalize γ. The last term is dependent on model bias Δ. Thus c_1 is roughly

$$c_1 = \text{Stability} + \text{Generalization} + \text{Model Bias}$$

To answer the question, 'How tight is the upper bound?', we use evaluation of curvature and memorization scores in Section 6.3 and find that the *linear relationship* from Equation 8 does hold true. We briefly and qualitatively discuss the validity of our assumptions in practical settings. Research (Hardt et al., 2016) has shown that using stochastic gradient methods (such as stochastic gradient descent) to train models attains small generalization error. Further, it has been shown that stochastic gradient is uniformly stable (Hardt et al., 2016). Thus the assumptions of stability (Equation 3) and generalization (Equation 4) are reasonable. Model bias is a property of the model, and a uniform bound across different datasets seems reasonable. And finally, the υ-adjacency can be ensured through construction. In practice, this might not be needed because the size of the ball $B_p(\upsilon)$ is unconstrained. Thus, two samples from the same class that are 'similar' may be sufficient to satisfy this requirement (note that this will result in a non-zero first term of Equation 9). With the size of modern datasets, this assumption is also reasonable.

Remark. Without assuming loss boundedness, we can state Theorem 5.1 for cross-entropy replacing L with 1, if

$$\forall h \in \mathrm{Range}(\mathcal{A}), \forall k \quad 0 < \Pr[h(x_k) = y_k] < 1.$$

Note the boundary condition that probability cannot be exactly 0 or 1. This is a reasonable assumption in a practical setting. The proof is provided in Appendix A.4. The main takeaway is that when the loss is bounded the expected difference in loss is the same as the memorization score, however when the loss is cross entropy the expected difference in loss upper bounds the memorization score.

5.2. Privacy and Input Loss Curvature

In this section, we present the second link between input loss curvature and privacy. To make the connection between input curvature and privacy we leverage stability. To establish the curvature-privacy link we first present Lemma 5.2 which links the stability constant and privacy. In doing so, we further improve the bounds in Wang et al. (2016), from $L(e^{\epsilon} - 1)$ to $L(1 - e^{-\epsilon})$.

Lemma 5.2 (Privacy $\implies$ Stability). *Assume boundedness of the loss, i.e., $0 \le \ell \le L$. Then, any ϵ-differential private algorithm satisfies $L(1 - e^{-\epsilon})$-stability.*

Sketch of Proof We start with the difference in the expected loss of adjacent datasets. Next, we assume that models resulting from training on S and $S^{\setminus i}$ for some i have distributions p and p', respectively. We use the properties of the expectation operator to expand the resultant terms. Next, by upper bounding the expectation using loss boundedness and performing some algebraic manipulations we arrive at the result. The full proof is provided in Appendix A.5.

Here we present our second main contribution in the form of Theorem 5.3 linking privacy and input loss curvature.

Theorem 5.3 (Privacy $\implies$ Low Input Loss Curvature). *Let $\mathcal{A}$ be a randomized algorithm which is ϵ-differentially private and the assumptions of error stability 3, generalization 4, and uniform model bias 5 hold. Further, assume $0 \le \ell \le L$. Then for two adjacent datasets $S, S^{\setminus i} \sim \mathcal{D}$ with a probability at least $1 - \delta$ we have*

$$\mathbb{E}_{z,\phi}[\mathrm{Curv}_{\phi}(z, S)] \le L(m+1)(1 - e^{-\epsilon}) + c_2 \tag{10}$$

$$c_2 = (4m-1)\gamma + 2(m-1)\Delta + \frac{\rho}{6}\mathbb{E}[\|\alpha\|^3] \tag{11}$$

Sketch of Proof Starting at Lemma A.4 we take the expectation over z, then we use the stability assumption. Rearranging the expressions and using Lemma 5.2 we arrive at the result. The full proof is provided in Appendix A.7.

Interpreting Theorem 5.3. Focusing on Equation 10, we see that a stronger privacy guarantee ensures reduced average input loss curvature. To validate the tightness of the bound we use evaluations of curvature and privacy in Section 6.4. Similar to Theorem 5.1, c_2 can be thought of as having two components, the generalization term γ and the model bias term Δ. By choosing α carefully (see previous discussion on Theorem 5.1) the last term can be ignored. Thus c_2 can be thought as c_2 = Generalization + Model Bias. The validity of our assumptions in practical settings is reasonable as previously discussed for Theorem 5.1.

5.3. Privacy and Memorization

The definition of stability-based memorization (Feldman, 2019) is very much related to privacy. Notably, Feldman (2019) explored this link, demonstrating that under specific conditions, algorithms that do not memorize cannot achieve optimal generalization performance. Feldman (2019) showed that this memorization-generalization result stems from the long-tailed nature of data. Our exploration in determining how memorization and privacy are related, is, however, different. In particular, we show that the memorization score is upper bounded by $1 - e^{-\epsilon}$ for any ϵ-DP algorithm. While this result is relatively straightforward, we state it for completeness as it is still a critical link in understanding memorization.

Theorem 5.4 (Privacy $\implies$ Less Memorization). *Let $\mathcal{A}$ be an ϵ-differentially private algorithm and z_i be the i^{th} element of $S \in \mathcal{Z}^m$. Then, we have*

$$\forall i \in \{1, \cdots, m\}, \quad \mathrm{mem}(\mathcal{A}, S, i) \le 1 - e^{-\epsilon}. \tag{12}$$

Sketch of Proof We start with the definition of ϵ-differential privacy, with simple algebraic manipulation, and repetitively using the definition of ϵ-differential privacy we arrive at the result. Note that this result can also be readily extended to (ϵ, δ_p)-differential privacy, i.e. Theorem 5.4 holds for an (ϵ, δ_p)-differential private algorithm with a probability $1 - \delta_p$. The full proof is provided in Appendix A.1.

6. Experiments

6.1. Experimental Setup

Datasets. To evaluate our theory we consider the classification task using standard vision datasets as the pre-computed stability-based memorization scores from Feldman & Zhang (2020) are available for CIFAR100 (Krizhevsky et al., 2009) and ImageNet (Russakovsky et al., 2015) datasets.

Architectures. For some experiments we train ResNet18 (He et al., 2016) models from scratch, while for others we use pre-trained Small Inception (Szegedy et al., 2015) and ResNet50 models released by Feldman & Zhang (2020). Details regarding the model used are specified at the beginning of each experiment section.

Training. For experiments that use private models, we use the Opacus library (Yousefpour et al., 2021) to train ResNet18 models for 20 epochs till the privacy budget is reached. We use DP-SGD (Abadi et al., 2016) with the maximum gradient norm set to 1.0 and privacy parameter $\delta = 1 \times 10^{-5}$. The initial learning rate was set to 0.001. The learning rate is decreased by 10 at epochs 12 and 16. When training on CIFAR10 and CIFAR100 datasets the batch size is set to 128. For both CIFAR10 and CIFAR100 datasets, we used the following sequence of data augmentations for training: resize (32×32), random crop, and random horizontal flip, this is followed by normalization.

Testing. During testing no augmentations were used, i.e. we used resize followed by normalization. When using pre-trained models from Feldman & Zhang (2020) we validated the accuracy of the models before performing experiments. To improve reproducibility, we have provided the code at this github link.

6.2. Estimating Input Loss Curvature

To corroborate the theoretical findings presented in the prior section, an efficient methodology for computing input loss curvature is needed as computing the full Hessian is computationally intensive. For this we assume H is positive semi-definite (see details in Appendix A.10). This lets us compute the trace of H, using Hutchinson's trace estimator (Hutchinson, 1989; Garg et al., 2023) from which we have

$$\operatorname{tr}(H) = \mathbb{E}_v\left[v^T H v\right], \tag{13}$$

where $v \in \mathbb{R}^d$ belongs to a Rademacher distribution. Using the finite step approximation similar to Moosavi-Dezfooli et al. (2019); Garg et al. (2023) and the symmetric nature of the Hessian we have

$$\operatorname{tr}(H) \leq \operatorname{tr}(H^2) = \frac{1}{n}\sum_{i=0}^{n} \|Hv_i\|_2^2$$
$$Hv \propto \frac{\partial\left(L(x+hv) - L(x)\right)}{\partial x}$$
$$\operatorname{tr}(H^2) \propto \frac{1}{n}\sum_{i=0}^{n} \left\| \frac{\partial\left(L(x+hv) - L(x)\right)}{\partial x} \right\|_2^2$$
$$\operatorname{Curv}(x) \propto \frac{1}{n}\sum_{i=0}^{n} \left\| \frac{\partial\left(L(x+hv) - L(x)\right)}{\partial x} \right\|_2^2, \tag{14}$$

where n is the number of Rademacher vectors to average. For all our experiments we used $h = 1 \times 10^{-3}$ and $n = 10$. We found the results to be robust to changes in h; we varied it from 1×10^{-1} to 1×10^{-3}. We also varied n from $5, 10, 20$ and found the results to be robust to changes in n .

6.3. Input Curvature and Memorization

In this section, we present results on CIFAR100 and ImageNet datasets for the first link between memorization and input loss curvature (Theorem 5.1).

Experiment. Here we aim to understand how memorization changes with curvature. The experiment aims to plot the memorization score vs. curvature score to validate our theoretical results. We calculate curvature scores by averaging over many seeds at the end of training. This measurement is proportional to the expected curvature score, i.e. $\mathbb{E}_\phi[\operatorname{Curv}_\phi(z_i, S^{\setminus i})]$ in Theorem 5.1.

For this experiment, we used 1000 models trained on CIFAR100 and 100 models trained on ImageNet obtained from Feldman & Zhang (2020)'s 0.7 subset ratio repository. We calculated the curvature score for each sample in the training set using Equation 14. We then compiled a dataset comprising each sample's memorization score and curvature score. Precomputed memorization scores were obtained from Feldman & Zhang (2020)'s repository. We averaged these scores across all models (1000 for CIFAR100 and 100 for ImageNet) to form an averaged dataset, which was divided into 50 bins based on memorization score. For example, bin 0 includes samples with memorization scores from 0 to 0.02 and the corresponding curvature scores, bin 1 includes samples in the memorization score range of 0.02 to 0.04, and so on. The average memorization score and maximum curvature score (since curvature is an upper bound) for each bin were used to create a scatter plot as shown in Figures 3(a) and 3(b). For CIFAR100, the Small Inception model was used, and for ImageNet, the ResNet50 model was used, both sourced from Feldman & Zhang (2020).

Results. We provide the results for CIFAR100 and ImageNet datasets in Figure 3(a) and 3(b) respectively. The figures also visualize the best-fit (shown in red) based on Theorem 5.1. From the results, we see a clear linear relation. The results from the experiment show that the curvature scores have a strong linear trend with respect to memorization, in line with Theorem 5.1.

Accounting for Variables in Practice. Notably, the linearity of the relation between memorization and curvature

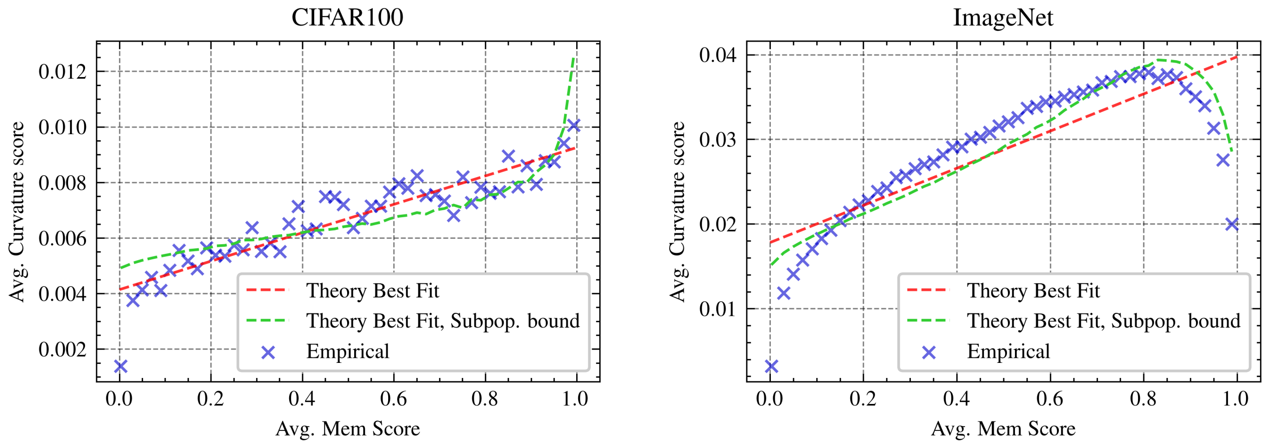


Figure 3. Plot of memorization score vs. input loss curvature at the end of training for CIFAR100 (average over 1000 Small Inception models) and ImageNet (average over 100 ResNet50) datasets.

diminishes at the extreme ends of the data range. This phenomenon is particularly pronounced in ImageNet results, as shown in Figure 3(b). This is because the loss bound L (refer to Equation 8) is not constant and the bound changes for each sub-population. Here, we can treat each memorization bin as a sub-population. Hence, when using cross entropy loss we found a better fit, if the loss bound is assumed, and the loss bound at convergence ismodeled. Accounting for the change in loss bound with sub-population size we see a much improved match. This is observed when comparing the best-fit results in green (assuming sub-population loss bound) vs red (no sub-population loss bound) in Figure 3.

To obtain an improved fit seen in Figure 3 we assumed the loss bound reduces in the square root of the sub-population size (Bousquet & Elisseeff, 2002). Since the theoretical curvature score from Equation 7 is proportional to the computed curvature score (Equation 14), we can rewrite Equation 8 from Theorm 5.1 using two parameters p_1, c_1 as

$$|\text{mem}(\mathcal{A}, S, i)| \leq \frac{p_1}{L} \cdot \mathbb{E}_\phi[\text{Curv}_\phi(z_i, S^{\setminus i})] + c_1$$

Using $L \propto m_{sub}^{-0.5}$, where m_{sub} is the number of samples in each sub-population we can model the relation as

$$|\text{mem}(\mathcal{A}, S, i)| \leq p_1 \cdot \sqrt{m_{sub}} \cdot \mathbb{E}_\phi[\text{Curv}_\phi(z_i, S^{\setminus i})] + c_1$$
$$s.t. \quad p_1, c_1 > 0. \tag{15}$$

Fitting parameters p_1, c_1 to the data results in the green plot in Figures 3(a) and 3(b), where we see much improved match between results and our theory. Thus, these results strongly agree with and validate Theorem 5.1.

6.4. Privacy and Input Loss Curvature

In this section, we present the results on CIFAR10 and CIFAR100 datasets to verify the link between privacy and input loss curvature (Theorem 5.3).

Experiment. To study the relation between privacy and curvature, we train private ResNet18 models on CIFAR10 and CIFAR100 using DP-SGD (Abadi et al., 2016) and calculate the curvature scores. We aim to plot privacy budget vs curvature score and validate Theorem 5.3. Specifically, we train models with privacy budgets ϵ ranging from 5 to 100, in increments of 5. We train 10 seeds for every privacy budget, and the curvature score is averaged over the 10 seeds and all the dataset samples.

Accounting for Variables in Practice. For our experiments, we use cross entropy trained private models, where the loss is unbounded. However, Theorem 5.3 assumes bounded loss. Thus, we obtain abound on the loss at convergence for each privacy budget. We model the loss bound as a function of privacy using $L(\epsilon) = a + be^{-c\epsilon}$. The fit of this model is shown in Figure 4. Using this loss bound model, Theorem 5.3 can be re-written as

$$\begin{aligned}\mathbb{E}_{z,\phi}[\text{Curv}_\phi(z, S)] &\leq L(\epsilon) \cdot (m+1) \cdot (1 - e^{-\epsilon}) + c_2 \\ &\leq (a + be^{-c\epsilon}) \cdot (m+1) \cdot (1 - e^{-\epsilon}) + c_2,\end{aligned} \tag{16}$$

where c_2 is treated as a constant when trying to fit the data to Equation 16. The data and the best fit model using Equation 16 are shown in Figure 5.

Results. The result of plotting the average convergence loss and privacy budget is shown in Figure 4 along with the best-fit model (in dashed blue line), demonstrating a strong match. Next, Figure 5 shows the result of studying the link between input loss curvature and privacy budget. The scatter plot shows curvature vs privacy. We visualize the data and the best fit (dashed line) using the model from Equation 16. Again, we see a very strong match. All these results strongly correlate with theory and validate Theorem 5.3.

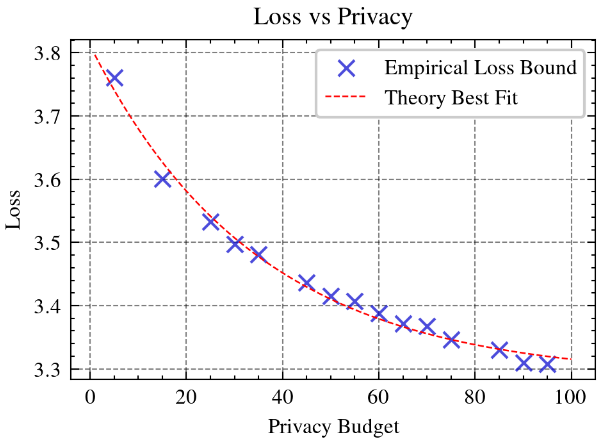


Figure 4. Plot of differential privacy vs loss bound for CIFAR100 trained with cross-entropy and the best fit curve (dashed).

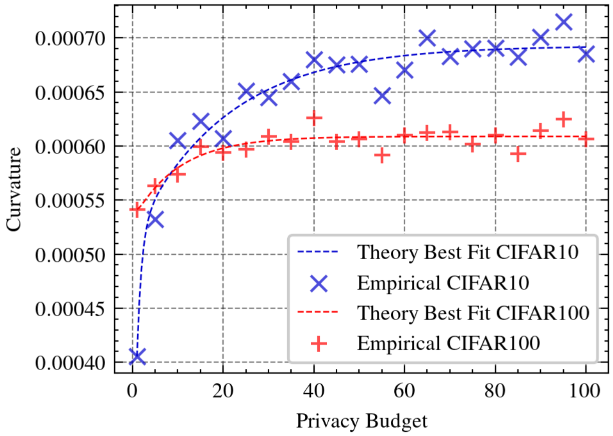


Figure 5. Plot of privacy vs loss curvature for CIFAR10 and CIFAR100. The best-fit curve (dashed) is predicted by Theorem 5.3.

6.5. Memorization and Privacy

In this section, we present the results for the link between memorization and privacy (Theorem 5.4).

Experiment. The goal of the study is to estimate the memorization score of samples when the models have differential privacy guarantees. Since Theorem 5.4 provides an upper bound, we are interested in how privacy affects most memorized examples. This enables us to reduce the compute requirement, and we consider the top 500 most memorized samples from CIFAR100 as reported in Feldman & Zhang (2020) and study how these scores change as privacy guarantees are varied. For this experiment we first split the CIFAR100 training set into two, set a contains all examples that are not the top 500 most memorized examples, and set b contains the top 500 most memorized examples as reported by Feldman & Zhang (2020). From b we randomly sample half the dataset called $b^{0.5}$. We concatenate a and $b^{0.5}$ to get our training set. This is used to train a ResNet18 model using DP-SGD (Abadi et al., 2016, Differentially Private SGD). We repeat the process of random sub-sampling of b and training for 40 seeds. By keeping track of what samples of b were present in each training run we can estimate the memorization score of the top 500 most memorized examples. This process is repeated 6 times for privacy budgets $\epsilon = 1, 10, 20, 30, 40, 50$ with $\delta = 1 \times 10^{-5}$ to train a total of 240 private models (previously described in Section 6.1).

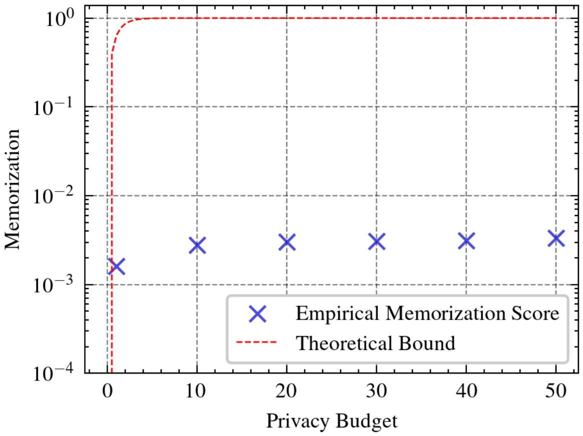


Figure 6. Plot of differential privacy vs memorization for CIFAR100 and the upper bound from the Theorem 5.4.

Results. The average memorization scores for the top 500 most memorized examples across various privacy budgets (ϵ) are presented in Figure 6. As a reference, we also plot the upper bound from Theorem 5.4 in the same plot. Note that Figure 6 is a semi-log plot. The results align with Theorem 5.4, showing an increase in memorization score as the privacy budget increases (i.e. privacy budget $\epsilon \uparrow$). Further, the memorization scores are significantly lower than the bound from Theorem 5.4 supporting Nasr et al. (2021)'s observation that DP-SGD may be overly conservative.

7. Conclusion

This paper explores the theoretical link between memorization, curvature, and privacy. Understanding this link is critical since input curvature offers ~ 3 orders of magnitude compute efficiency when calculating memorization scores. The theoretical analysis relies on three assumptions, stability, generalization, and Lipshitzness, and thus can be applied in non-convex settings such as DNNs. Our main result shows that memorization is upper-bounded by the curvature of the loss with respect to input and privacy. Further, we presented two theorems that complete the links between memorization, privacy, and input loss curvature. To test the theory we use standard DNNs for image classification using CIFAR100 and ImageNet datasets. Our results show a very strong match between our theoretical findings andresults. Results in this paper provide evidence for the link between memorization, input loss curvature, and privacy strengthening the understanding of DNNs and their properties.

Acknowledgment

This work was supported in part by, the Center for the Co-Design of Cognitive Systems (CoCoSys), a DARPA-sponsored JUMP 2.0 center, the Semiconductor Research Corporation (SRC), and the National Science Foundation.

Impact Statement

The research presented in this paper fills significant gaps in our understanding of DNNs. We focused on the relationship between memorization, input loss curvature, and privacy. This finding is key for various applications, as it provides a clearer framework for leveraging the significant ($\sim$ 3 orders of magnitude) efficiencies in computing memorization scores when using input loss curvature. This work, therefore, not only develops our theoretical understanding of DNNs but also offers practical insights for developing more effective machine learning and deep learning models and algorithms.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Andriushchenko, M. and Flammarion, N. Towards understanding sharpness-aware minimization. In *International Conference on Machine Learning*, pp. 639–668. PMLR, 2022.

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pp. 233–242. PMLR, 2017.

Bae, J., Ng, N., Lo, A., Ghassemi, M., and Grosse, R. B. If influence functions are the answer, then what is the question? *Advances in Neural Information Processing Systems*, 35:17953–17967, 2022.

Basu, S., Pope, P., and Feizi, S. Influence functions in deep learning are fragile. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=xHKVVHGDOEk`.

Bousquet, O. and Elisseeff, A. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

Brown, G., Bun, M., Feldman, V., Smith, A., and Talwar, K. When is memorization of irrelevant training data necessary for high-accuracy learning? In *Proceedings of the 53rd annual ACM SIGACT symposium on theory of computing*, pp. 123–132, 2021.

Carlini, N., Erlingsson, U., and Papernot, N. Distribution density, tails, and outliers in machine learning: Metrics and applications. *arXiv preprint arXiv:1910.13427*, 2019.

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

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pp. 265–284. Springer, 2006.

Fawzi, A., Moosavi-Dezfooli, S.-M., Frossard, P., and Soatto, S. Empirical study of the topology and geometry of deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3762–3770, 2018.

Feldman, V. Does learning require memorization? a short tale about a long tail. *arXiv preprint arXiv:1906.05271*, 2019.

Feldman, V. and Vondrak, J. High probability generalization bounds for uniformly stable algorithms with nearly optimal rate. In *Conference on Learning Theory*, pp. 1270–1279. PMLR, 2019.

Feldman, V. and Zhang, C. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=6Tm1mposlrM`.

Garg, I. and Roy, K. Samples with low loss curvature improve data efficiency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20290–20300, 2023.

Garg, I., Ravikumar, D., and Roy, K. Memorization through the lens of curvature of loss function around samples. *arXiv preprint arXiv:2307.05831*, 2023.

Hardt, M., Recht, B., and Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pp. 1225–1234. PMLR, 2016.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hutchinson, M. F. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.

Jiang*, Y., Neyshabur*, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJgIPJBFvH.

Jiang, Z., Zhang, C., Talwar, K., and Mozer, M. C. Characterizing structural regularities of labeled data in overparameterized models. *arXiv preprint arXiv:2002.03206*, 2020.

Kearns, M. and Ron, D. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. In *Proceedings of the tenth annual conference on Computational learning theory*, pp. 152–162, 1997.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=H1oyRlYgg.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images, 2009.

Kwon, J., Kim, J., Park, H., and Choi, I. K. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, pp. 5905–5914. PMLR, 2021.

Liu, S., Niles-Weed, J., Razavian, N., and Fernandez-Granda, C. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33:20331–20342, 2020.

Lukasik, M., Nagarajan, V., Rawat, A. S., Menon, A. K., and Kumar, S. What do larger image classifiers memorise? *arXiv preprint arXiv:2310.05337*, 2023.

Maini, P., Garg, S., Lipton, Z., and Kolter, J. Z. Characterizing datapoints via second-split forgetting. *Advances in Neural Information Processing Systems*, 35:30044–30057, 2022.

Moosavi-Dezfooli, S.-M., Fawzi, A., Uesato, J., and Frossard, P. Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9078–9086, 2019.

Murakonda, S. K. and Shokri, R. Ml privacy meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning. *arXiv preprint arXiv:2007.09339*, 2020.

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

Nasr, M., Songi, S., Thakurta, A., Papernot, N., and Carlin, N. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on security and privacy (SP)*, pp. 866–882. IEEE, 2021.

Nesterov, Y. and Polyak, B. T. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Pleiss, G., Zhang, T., Elenberg, E., and Weinberger, K. Q. Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems*, 33:17044–17056, 2020.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Schioppa, A., Zablotskaia, P., Vilar, D., and Sokolov, A. Scaling up influence functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8179–8186, 2022.

Schioppa, A., Filippova, K., Titov, I., and Zablotskaia, P. Theoretical and practical perspectives on what influence functions do. *arXiv preprint arXiv:2305.16971*, 2023.

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

Stock, P. and Cisse, M. Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 498–512, 2018.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Toneva, M., Sordoni, A., des Combes, R. T., Trischler, A., Bengio, Y., and Gordon, G. J. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJlxm30cKm.

Wang, Y.-X., Lei, J., and Fienberg, S. E. Learning with differential privacy: Stability, learnability and the sufficiency and necessity of erm principle. *The Journal of Machine Learning Research*, 17(1):6353–6392, 2016.

Wu, D., Xia, S.-T., and Wang, Y. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020.

Yousefpour, A., Shilov, I., Sablayrolles, A., Testuggine, D., Prasad, K., Malek, M., Nguyen, J., Ghosh, S., Bharadwaj, A., Zhao, J., Cormode, G., and Mironov, I. Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298*, 2021.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

A. Proofs

A.1. Proof of Theorem 5.4

Consider $S, S^{\setminus i}$ from construction we have $||S - S^{\setminus i}|| = 1$. Next let $\mathcal{R} \subset \text{Range}(\mathcal{A})$ such that $\mathcal{R} = \{h \mid h(x_i) = y_i\}$. Since $\mathcal{A}$ is ϵ-differentially private then it follows from the definition of differential privacy in Equation 1 that

$$\Pr_{\phi}[h_S^{\phi} \in \mathcal{R}] \leq e^{\epsilon} \Pr_{\phi}[h_{S\setminus i}^{\phi} \in \mathcal{R}] \tag{17}$$

Since $\mathcal{R} = \{h \mid h(x_i) = y_i\}$ we have

$$\Pr_{\phi}[h_S^{\phi} \in \mathcal{R}] = \Pr_{\phi}[h_S^{\phi}(x_i) = y_i] \tag{18}$$

Using Equations 17 and 18 we have

$$\begin{aligned}
\Pr_{\phi}[h_S^{\phi}(x_i) = y_i] &\leq e^{\epsilon} \Pr_{\phi}[h_{S^{\setminus i}}^{\phi}(x_i) = y_i] \qquad (19)\\
\Pr_{\phi}[h_S^{\phi}(x_i) = y_i] &\leq e^{\epsilon} \Pr_{\phi}[h_{S^{\setminus i}}^{\phi}(x_i) = y_i] \pm \Pr_{\phi}[h_{S^{\setminus i}}^{\phi}(x_i) = y_i]\\
\Pr_{\phi}[h_S^{\phi}(x_i) = y_i] - \Pr_{\phi}[h_{S^{\setminus i}}^{\phi}(x_i) = y_i] &\leq e^{\epsilon} \Pr_{\phi}[h_{S^{\setminus i}}^{\phi}(x_i) = y_i] - \Pr_{\phi}[h_{S^{\setminus i}}^{\phi}(x_i) = y_i]\\
\Pr_{\phi}[h_S^{\phi}(x_i) = y_i] - \Pr_{\phi}[h_{S^{\setminus i}}^{\phi}(x_i) = y_i] &\leq (e^{\epsilon} - 1) \Pr_{\phi}[h_{S^{\setminus i}}^{\phi}(x_i) = y_i]\\
\text{mem}(\mathcal{A}, S, i) &\leq (e^{\epsilon} - 1) \Pr_{\phi}[h_{S^{\setminus i}}^{\phi}(x_i) = y_i]
\end{aligned}$$

Using Equation 19, we have the lower bound on $\Pr_{\phi}[h_{S^{\setminus i}}^{\phi}(x_i) = y_i]$ as

$$\Pr_{\phi}[h_{S^{\setminus i}}^{\phi}(x_i) = y_i] \geq e^{-\epsilon} \Pr_{\phi}[h_S^{\phi}(x_i) = y_i]$$

Thus we have

$$\begin{aligned}
\text{mem}(\mathcal{A}, S, i) &\leq (e^{\epsilon} - 1) e^{-\epsilon} \Pr_{\phi}[h_S^{\phi}(x_i) = y_i]\\
\text{mem}(\mathcal{A}, S, i) &\leq (1 - e^{-\epsilon}) \Pr_{\phi}[h_S^{\phi}(x_i) = y_i]
\end{aligned}$$

Since $\sup \Pr_{\phi}[h_S^{\phi}(x_i) = y_i] = 1$ we have the result

$$\text{mem}(\mathcal{A}, S, i) \leq 1 - e^{-\epsilon} \quad \blacksquare$$

A.2. Proof of Lemma A.2

Lemma A.1. *If the generalization assumption 4 holds then we know here exists γ such that with probability $1 - \delta$*

$$\begin{aligned}
\mathbb{E}_{\phi}[|R_{emp}(h_{S^{\setminus i}}^{\phi}, S) - R(h_{S^{\setminus i}}^{\phi})|] &\leq \gamma \qquad (20)\\
\mathbb{E}_{\phi}[|R_{emp}(h_S^{\phi}, S) - R(h_S^{\phi})|] &\leq \gamma \qquad (21)\\
\mathbb{E}_{\phi}[|R_{emp}(h_S^{\phi}, S^{\setminus i}) - R(h_S^{\phi})|] &\leq \gamma \qquad (22)
\end{aligned}$$

Proof of Lemma A.1 From (Feldman & Vondrak, 2019) we know that with a confidence δ we have

$$\Pr_{S \sim \mathcal{D}^m}\left[|R_{emp}(h, S) - R(h)| \geq c\left(\beta' \ln(m) \ln(m/\delta) + \frac{\sqrt{\ln(1/\delta)}}{\sqrt{m}}\right)\right] \leq \delta$$

Where β' is the uniform stability bound. Thus with a confidence of at least $1 - \delta$ we can say:

$$|R_{emp}(h, S) - R(h)| < c\left(\beta' \ln(m) \ln(m/\delta) + \frac{\sqrt{\ln(1/\delta)}}{\sqrt{m}}\right)$$

Thus if we set

$$\gamma'(m) = c\left(\beta' \ln(m)\ln(m/\delta) + \frac{\sqrt{\ln(1/\delta)}}{\sqrt{m}}\right)$$

we have

$$|R_{emp}(h, S) - R(h)| < \gamma'(m) \tag{23}$$

Thus as a direct consequence of Equation 23 we can say

$$\forall S, S^{\setminus i} \sim \mathcal{D}^m,\ \mathbb{E}_\phi[|R_{emp}(h^\phi_{S^{\setminus i}}, S^{\setminus i}) - R(h^\phi_{S^{\setminus i}})|]\ \leq \gamma'(m-1) \tag{24}$$

$$\forall S \sim \mathcal{D}^m,\ \mathbb{E}_\phi[|R_{emp}(h^\phi_S, S) - R(h^\phi_S)|]\ \leq \gamma'(m) \tag{25}$$

$$\begin{aligned}
\mathbb{E}_\phi[|R_{emp}(h^\phi_{S^{\setminus i}}, S) - R(h^\phi_{S^{\setminus i}})|] &= \mathbb{E}_\phi\left[\left|\frac{1}{m}\ell\left(h^\phi_{S^{\setminus i}}, z_i\right)\right|\right] + \mathbb{E}_\phi\left[\left|\frac{m-1}{m}R_{emp}(h^\phi_{S^{\setminus i}}, S^{\setminus i}) - R(h^\phi_{S^{\setminus i}})\right|\right] \\
&= \frac{1}{m}\mathbb{E}_\phi\left[\left|\ell\left(h^\phi_{S^{\setminus i}}, z_i\right)\right|\right] + \mathbb{E}_\phi\left[\left|\frac{m-1}{m}R_{emp}(h^\phi_{S^{\setminus i}}, S^{\setminus i}) - \frac{m-1}{m}R(h^\phi_{S^{\setminus i}}) - \frac{1}{m}R(h^\phi_{S^{\setminus i}})\right|\right] \\
&\leq \frac{L}{m} + \mathbb{E}_\phi\left[\left|\frac{m-1}{m}R_{emp}(h^\phi_{S^{\setminus i}}, S^{\setminus i}) - \frac{m-1}{m}R(h^\phi_{S^{\setminus i}}) - \frac{1}{m}R(h^\phi_{S^{\setminus i}})\right|\right] \\
&\leq \frac{L}{m} + \mathbb{E}_\phi\left[\left|\frac{m-1}{m}R_{emp}(h^\phi_{S^{\setminus i}}, S^{\setminus i}) - \frac{m-1}{m}R(h^\phi_{S^{\setminus i}}) - \frac{1}{m}R(h^\phi_{S^{\setminus i}}) \pm \frac{1}{m}R(h^*)\right|\right] \\
&\leq \frac{L}{m} + \mathbb{E}_\phi\left[\left|\frac{m-1}{m}R_{emp}(h^\phi_{S^{\setminus i}}, S^{\setminus i}) - \frac{m-1}{m}R(h^\phi_{S^{\setminus i}}) - \frac{1}{m}R(h^*)\right|\right] + \Delta \\
&\leq \frac{L}{m} + \mathbb{E}_\phi\left[\left|\frac{m-1}{m}R_{emp}(h^\phi_{S^{\setminus i}}, S^{\setminus i}) - \frac{m-1}{m}R(h^\phi_{S^{\setminus i}}) - \frac{1}{m}R(h^*)\right|\right] + \Delta \\
&\leq \frac{L}{m} + \left|\frac{m-1}{m}\gamma'(m-1)\right| + \left|\frac{1}{m}R(h^*)\right| + \Delta \\
&\leq \frac{L}{m} + \left|\frac{m-1}{m}\gamma'(m-1)\right| + \frac{L}{m} + \Delta \\
&\leq \frac{2L}{m} + \frac{m-1}{m}\gamma'(m-1) + \Delta
\end{aligned}$$

Now consider

$$\begin{aligned}
R_{emp}(h^\phi_S, S^{\setminus i}) - R(h^\phi_S) &= R_{emp}(h^\phi_S, S^{\setminus i}) - R(h^\phi_S) \\
&= \frac{1}{m-1}\sum_{j=1, j\neq i}^{m} \ell(h^\phi_S, z_j) - R(h^\phi_S) \\
&= \frac{1}{m-1}\sum_{j=1}^{m} \ell(h^\phi_S, z_j) - \frac{1}{m-1}\ell(h^\phi_S, z_i) - R(h^\phi_S) \\
&\leq \frac{m}{m-1}R_{emp}(h^\phi_S, S) - R(h^\phi_S) \\
&\leq \gamma'(m) + \frac{1}{m-1}R_{emp}(h^\phi_S, S) \\
&\leq \gamma'(m) + \frac{1}{m-1}L \\
|R_{emp}(h^\phi_S, S^{\setminus i}) - R(h^\phi_S)| &\leq \gamma'(m) + \frac{L}{m-1}
\end{aligned}$$

Thus if we set $\gamma(m) = \max\left\{\frac{2L}{m} + \frac{m-1}{m}\gamma'(m-1) + \Delta, \gamma'(m) + \frac{L}{m-1}\right\}$ we get $\forall S, S^{\setminus i} \sim \mathcal{D}^m$

$$\mathbb{E}_\phi[|R_{emp}(h^\phi_{S^{\setminus i}}, S) - R(h^\phi_{S^{\setminus i}})|] \leq \gamma \tag{26}$$

$$\mathbb{E}_\phi[|R_{emp}(h^\phi_S, S) - R(h^\phi_S)|] \leq \gamma \tag{27}$$

$$\mathbb{E}_\phi[|R_{emp}(h^\phi_S, S^{\setminus i}) - R(h^\phi_S)|] \leq \gamma \tag{28}$$

Using error stability from assumption (see Equation 3) introduced by Kearns & Ron (1997) without loss of generality, we can write

$$\mathbb{E}_{\phi,z\sim\mathcal{D}}[\ell(h^\phi_S, z)] - \mathbb{E}_{\phi,z\sim\mathcal{D}}[\ell(h^\phi_{S^{\setminus i}}, z)] \leq \beta$$
$$\mathbb{E}_\phi[R(h^\phi_S) - R(h^\phi_{S^{\setminus i}})] \leq \beta$$

Lemma A.2. *If assumptions of error stability 3, generalization 4, and uniform model bias 5 hold, then for all i, j and two adjacent datasets $S, S^{\setminus i} \sim \mathcal{D}$ with a probability at least $1-\delta$ it holds that*

$$\begin{aligned}\left|\mathbb{E}_\phi[\ell(h^\phi_S, z_i) - \ell(h^\phi_{S^{\setminus i}}, z_j)]\right| &\leq m\beta + (4m-1)\gamma \\ &+ 2(m-1)\Delta.\end{aligned} \tag{29}$$

Lemma A.2 provides an upper bound on the expected loss difference between two adjacent datasets evaluated at any data point in the training set.

Proof of Lemma A.2 Using Lemma A.1 we know here exists γ such that with probability $1-\delta$

$$\mathbb{E}_\phi[|R_{emp}(h^\phi_{S^{\setminus i}}, S) - R(h^\phi_{S^{\setminus i}})|] \leq \gamma$$
$$\mathbb{E}_\phi[|R_{emp}(h^\phi_S, S) - R(h^\phi_S)|] \leq \gamma$$
$$\mathbb{E}_\phi[|R_{emp}(h^\phi_S, S^{\setminus i}) - R(h^\phi_S)|] \leq \gamma$$

Using Equations 20 and 21 we can upper bound the expected difference in empirical risk of adjacent datasets as

$$\mathbb{E}_\phi[R_{emp}(h^\phi_S, S) - R_{emp}(h^\phi_{S^{\setminus i}}, S)] \leq \beta + 2\gamma$$
$$\mathbb{E}_\phi\left[\frac{1}{m}\ell(h^\phi_S, z_i) + \frac{m-1}{m}R_{emp}(h^\phi_S, S^{\setminus i}) - \frac{1}{m}\ell(h^\phi_{S^{\setminus i}}, z_j) - \frac{m-1}{m}R_{emp}(h^\phi_{S^{\setminus i}}, S^{\setminus i})\right] \leq \beta + 2\gamma$$

$$\frac{1}{m}\mathbb{E}_\phi[\ell(h^\phi_S, z_i)] - \frac{1}{m}\mathbb{E}_\phi[\ell(h^\phi_{S^{\setminus i}}, z_j)] \leq \beta + 2\gamma + \frac{m-1}{m}\mathbb{E}_\phi[R_{emp}(h^\phi_{S^{\setminus i}}, S^{\setminus i}) - R_{emp}(h^\phi_S, S^{\setminus i})]$$
$$\mathbb{E}_\phi[\ell(h^\phi_S, z_i)] - \mathbb{E}_\phi[\ell(h^\phi_{S^{\setminus i}}, z_j)] \leq m\beta + 2m\gamma + (m-1)\,\mathbb{E}_\phi[R_{emp}(h^\phi_{S^{\setminus i}}, S^{\setminus i}) - R_{emp}(h^\phi_S, S^{\setminus i})]$$

We obtain the upper and lower bound for empirical risk using Equations 20 and 22 to get

$$\mathbb{E}_\phi[\ell(h^\phi_S, z_i)] - \mathbb{E}_\phi[\ell(h^\phi_{S^{\setminus i}}, z_j)] \leq m\beta + 2m\gamma + (m-1)\,\mathbb{E}_\phi[R(h^\phi_{S^{\setminus i}}) + \gamma - R(h^\phi_S) + \gamma]$$
$$\mathbb{E}_\phi[\ell(h^\phi_S, z_i) - \ell(h^\phi_{S^{\setminus i}}, z_j)] \leq m\beta + (4m-1)\gamma + (m-1)\,\mathbb{E}_\phi[R(h^\phi_{S^{\setminus i}}) - R(h^\phi_S)]$$

We add an subtract the risk of $h^* = \mathbb{E}[y|x]$ which is the true conditional of $\mathcal{D}^m$

$$\mathbb{E}_\phi[\ell(h^\phi_S, z_i) - \ell(h^\phi_{S^{\setminus i}}, z_j)] \leq m\beta + (4m-1)\gamma + (m-1)\,\mathbb{E}_\phi[R(h^\phi_{S^{\setminus i}}) - R(h^\phi_S) \pm R(h^*)]$$
$$\mathbb{E}_\phi[\ell(h^\phi_S, z_i) - \ell(h^\phi_{S^{\setminus i}}, z_j)] \leq m\beta + (4m-1)\gamma + (m-1)\,\mathbb{E}_\phi[R(h^\phi_{S^{\setminus i}}) - R(h^*) - (R(h^\phi_S) - R(h^*))]$$

Using the uniform model bias bound from assumption 5 we have

$$\mathbb{E}_\phi[R(h^\phi_{S^{\setminus i}}) - R(h^*)] \leq \Delta$$
$$\mathbb{E}_\phi[R(h^\phi_S) - R(h^*)] \geq -\Delta$$

Hence we get

$$\mathbb{E}_\phi[\ell(h_S^\phi, z_i) - \ell(h_{S^{\setminus i}}^\phi, z_j)] \leq m\beta + (4m-1)\gamma + (m-1)\left[\Delta - (-\Delta)\right]$$
$$\mathbb{E}_\phi[\ell(h_S^\phi, z_i) - \ell(h_{S^{\setminus i}}^\phi, z_j)] \leq m\beta + (4m-1)\gamma + 2(m-1)\Delta$$

Since we can interchange $\ell(h_S^\phi, z_i)$ and $\ell(h_{S^{\setminus i}}^\phi, z_j)$ i.e. start with $\mathbb{E}_{z\sim\mathcal{D}}[\ell(h_{S^{\setminus i}}^\phi, z)] - \mathbb{E}_{z\sim\mathcal{D}}[\ell(h_S^\phi, z)]$ we have the result

$$\left|\mathbb{E}_\phi[\ell(h_S^\phi, z_i) - \ell(h_{S^{\setminus i}}^\phi, z_j)]\right| \leq m\beta + (4m-1)\gamma + 2(m-1)\Delta \quad \blacksquare$$

Lemma A.3. *If Lipschitz assumption 6 on the Hessian of ℓ holds from Nesterov & Polyak (2006) we have*

$$|\ell(h, z_1) - \ell(h, z_2) - \langle \nabla \ell(h, z_2), z_1 - z_2\rangle - \langle \nabla^2 \ell(h, z_2)(z_1 - z_2), z_1 - z_2\rangle| \leq \frac{\rho}{6}|z_1 - z_2|^3 \tag{30}$$

A.3. Proof of Theorem 5.1

From Lemma A.3 we have

$$-\frac{\rho}{6}|z_1 - z_2|^3 \leq \ell(h, z_1) - \ell(h, z_2) - \langle \nabla \ell(h, z_2), z_1 - z_2\rangle - \langle \nabla^2 \ell(h, z_2)(z_1 - z_2), z_1 - z_2\rangle \leq \frac{\rho}{6}|z_1 - z_2|^3$$

This gives us an upper bound on $\ell(h, z_1)$

$$\ell(h, z_1) \leq \frac{\rho}{6}|z_1 - z_2|^3 + \ell(h, z_2) + \langle \nabla \ell(h, z_2), z_1 - z_2\rangle + \langle \nabla^2 \ell(h, z_2)(z_1 - z_2), z_1 - z_2\rangle \tag{31}$$

Consider $z_j \in S$ such that $z_j = z_i + \alpha$ for some $j \neq i$ where $\alpha \in B_p(\upsilon)$ such that $\mathbb{E}[\alpha] = 0$ and $\mathbb{E}[\alpha^T \alpha] = 1$.

Without loss of generality from Lemma A.2 we have:

$$\mathbb{E}_\phi[\ell(h_S^\phi, z_i) - \ell(h_{S^{\setminus i}}^\phi, z_j)] \leq m\beta + (4m-1)\gamma + 2(m-1)\Delta$$

Using the upper bound from Equation 31, setting $z_1 = z_j, z_2 = z_i$ we have

$$\mathbb{E}_\phi[\ell(h_S^\phi, z_i) - \ell(h_{S^{\setminus i}}^\phi, z_j) - \frac{\rho}{6}\|\alpha\|^3 - \langle \nabla \ell(h_{S^{\setminus i}}^\phi, z_i), \alpha\rangle - \langle \nabla^2 \ell(h_{S^{\setminus i}}^\phi, z_i)\alpha, \alpha\rangle] \leq m\beta + (4m-1)\gamma + 2(m-1)\Delta$$
$$\mathbb{E}_\phi[\ell(h_S^\phi, z_i) - \ell(h_{S^{\setminus i}}^\phi, z_i) - \frac{\rho}{6}\|\alpha\|^3 - \langle \nabla \ell(h_{S^{\setminus i}}^\phi, z_i), \alpha\rangle - \alpha^T H^T \alpha] \leq m\beta + (4m-1)\gamma + 2(m-1)\Delta$$

Where $H = \nabla^2 \ell(h_{S^{\setminus i}}^\phi, z_i)$. Next, we take expectation over α we get

$$\mathbb{E}_\phi[\ell(h_S^\phi, z_i)] - \mathbb{E}_\phi[\ell(h_{S^{\setminus i}}^\phi, z_i)] - \frac{\rho}{6}\mathbb{E}_\alpha[\|\alpha\|^3] - \mathbb{E}_{\phi,\alpha}[\alpha^T H^T \alpha] \leq m\beta + (4m-1)\gamma + 2(m-1)\Delta$$
$$\mathbb{E}_\phi[\ell(h_S^\phi, z_i)] - \mathbb{E}_\phi[\ell(h_{S^{\setminus i}}^\phi, z_i)] - \frac{\rho}{6}\mathbb{E}_\alpha[\|\alpha\|^3] - \mathbb{E}_{\phi,\alpha}[\mathrm{tr}(H^T\, \mathbb{E}_\alpha[\alpha^T \alpha])] \leq m\beta + (4m-1)\gamma + 2(m-1)\Delta$$
$$\mathbb{E}_\phi[\ell(h_S^\phi, z_i)] - \mathbb{E}_\phi[\ell(h_{S^{\setminus i}}^\phi, z_i)] \leq \frac{\rho}{6}\mathbb{E}_\alpha[\|\alpha\|^3] + \mathbb{E}_\phi[\mathrm{tr}(H^T\, \mathbb{E}_\alpha[\alpha^T \alpha])] + m\beta + (4m-1)\gamma + 2(m-1)\Delta$$
$$\mathbb{E}_\phi[\ell(h_S^\phi, z_i)] - \mathbb{E}_\phi[\ell(h_{S^{\setminus i}}^\phi, z_i)] \leq \frac{\rho}{6}\mathbb{E}_\alpha[\|\alpha\|^3] + \mathbb{E}_\phi[\mathrm{tr}(H)] + m\beta + (4m-1)\gamma + 2(m-1)\Delta$$
$$\mathbb{E}_\phi[\ell(h_S^\phi, z_i)] - \mathbb{E}_\phi[\ell(h_{S^{\setminus i}}^\phi, z_i)] \leq \frac{\rho}{6}\mathbb{E}_\alpha[\|\alpha\|^3] + \mathbb{E}_\phi[\mathrm{tr}(\nabla^2 \ell(h_{S^{\setminus i}}^\phi, z_i))] + m\beta + (4m-1)\gamma + 2(m-1)\Delta$$

If we have $0 \leq \ell \leq L$ then:

$$\frac{1}{L}\left[\mathbb{E}_\phi[\ell(h_S^\phi, z_i)] - \mathbb{E}_\phi[\ell(h_{S^{\setminus i}}^\phi, z_i)]\right] = \mathrm{mem}(\mathcal{A}, S, i) \tag{32}$$

Since we can exchange S and $S^{\setminus i}$ Hence we have the result

$$|\mathrm{mem}(\mathcal{A}, S, i)| \leq \frac{\rho}{6L}\mathbb{E}_\alpha[\|\alpha\|^3] + \frac{1}{L}\mathbb{E}_\phi[\mathrm{tr}(\nabla^2 \ell(h_{S^{\setminus i}}^\phi, z_i))] + \frac{m\beta}{L} + \frac{(4m-1)\gamma}{L} + \frac{2(m-1)\Delta}{L} \quad \blacksquare$$

A.4. Proof of Theorem 5.1 for Cross-Entropy

For classification with one-hot ground truth labels we have cross entropy we have.

$$\begin{aligned}
\ell(h_S^\phi, z_i) &= -\ln(\Pr[h_S^\phi(x_i) = y_i]) \\
\ell(h_S^\phi, z_i) - \ell(h_{S^{\setminus i}}^\phi, z_j) &= \ln\left(\frac{\Pr[h_{S^{\setminus i}}^\phi(x_j) = y_j]}{\Pr[h_S^\phi(x_i) = y_i]}\right) \\
&= \ln\left(\frac{a}{b}\right)
\end{aligned}$$

$$\text{for} \quad \theta > -1 \quad \text{we have,} \quad \frac{\theta}{\theta+1} \le \ln(1+\theta)$$

$$\frac{\frac{a}{b} - 1}{\frac{a}{b}} \le \ln\left(\frac{a}{b}\right)$$

$$\frac{a-b}{a} \le \ln\left(\frac{a}{b}\right)$$

$$a - b \le \frac{a-b}{a} \le \ln\left(\frac{a}{b}\right) \quad \text{For } 0 < a \le 1$$

Thus we have

$$\Pr[h_{S^{\setminus i}}^\phi(x_j) = y_j] - \Pr[h_S^\phi(x_i) = y_i] \le \ell(h_S^\phi, z_i) - \ell(h_{S^{\setminus i}}^\phi, z_j)$$

Taking expectation over the randomness of $\mathcal{A}$ we have

$$\begin{aligned}
\mathbb{E}_\phi[\Pr[h_{S^{\setminus i}}^\phi(x_j) = y_j]] - \mathbb{E}_\phi[\Pr[h_S^\phi(x_i) = y_i]] &\le \mathbb{E}_\phi[\ell(h_S^\phi, z_i)] - \mathbb{E}_\phi[\ell(h_{S^{\setminus i}}^\phi, z_j)] \\
\text{mem}(\mathcal{A}, S, i) &\le \mathbb{E}_\phi[\ell(h_S^\phi, z_i)] - \mathbb{E}_\phi[\ell(h_{S^{\setminus i}}^\phi, z_j)] \quad \blacksquare
\end{aligned}$$

A.5. Proof of Lemma 5.2

Let $h \sim \mathcal{A}(\phi, S)$ have a pdf defined as $p(h)$, and $h' \sim \mathcal{A}(\phi, S^{\setminus i})$ have a pdf defined as $p'(h')$

$$
\begin{aligned}
\left|\mathbb{E}_{\phi,z}[\ell(h_S^{\phi}, z)] - \mathbb{E}_{\phi,z}[\ell(h_{S^{\setminus i}}^{\phi}, z)]\right| &= \left|\mathbb{E}_{z,\phi}[\ell(h_S^{\phi}, z)] - \mathbb{E}_{z,\phi}[\ell(h_{S^{\setminus i}}^{\phi}, z)]\right| \\
&= \left|\mathbb{E}_{z,\phi}[\ell(\mathcal{A}(\phi, S), z)] - \mathbb{E}_{z,\phi}[\ell(\mathcal{A}(\phi, S^{\setminus i}), z)]\right| \\
&= |\mathbb{E}_{z,h}[\ell(h, z)] - \mathbb{E}_{z,h'}[\ell(h', z)]| \\
&= \left|\int_z \int_h \ell(h, z) p(h) dh\, p(z) dz - \int_z \int_{h'} \ell(h', z) p'(h') dh'\, p(z) dz\right| \\
&= \left|\int_z \int_h \ell(h, z) p(h) dh\, p(z) dz - \int_z \int_h \ell(h, z) p'(h) dh\, p(z) dz\right| \\
&= \left|\int_z \int_h \ell(h, z)(p(h) - p'(h)) dh\, p(z) dz\right| \\
&\leq \left|\int_z \sup_h \ell(h, z) \int_{h: p(h) \geq p'(h)} (p(h) - p'(h)) dh\, p(z) dz\right| \\
&\leq \left|\sup_{h,z} \ell(h, z) \int_z p(z) dz \int_{h: p(h) \geq p'(h)} (p(h) - p'(h)) dh\right| \\
&\leq \left|L \int_{h: p(h) \geq p'(h)} p(h) - p'(h) dh\right| \\
&\leq \left|L \int_{h: p(h) \geq p'(h)} p(h) \left(1 - \frac{p'(h)}{p(h)}\right) dh\right| \\
&\leq \left|L \int_{h: p(h) \geq p'(h)} p(h) \left(1 - e^{-\epsilon}\right) dh\right| \\
&\leq \left|L(1 - e^{-\epsilon}) \int_{h: p(h) \geq p'(h)} p(h) dh\right| \\
&\leq \left|L(1 - e^{-\epsilon})\right| \\
&\leq L(1 - e^{-\epsilon}) \quad \blacksquare
\end{aligned}
$$

Lemma A.4. *If the assumptions of error stability 3, generalization 4, and uniform model bias 5 hold, then for two adjacent datasets $S, S^{\setminus i} \sim \mathcal{D}$ and for any $i, j \in \{1, \cdots, m\}$ with a probability at least $1 - \delta$ we have*

$$
\begin{aligned}
\mathbb{E}_{\phi}[\mathrm{tr}(\nabla^2 \ell(h_S^{\phi}, z_i))] \leq m\beta &+ (4m - 1)\gamma \\
&+ 2(m - 1)\Delta + \frac{\rho}{6}\mathbb{E}[\|\alpha\|^3] \\
&+ \mathbb{E}_{\phi}[\ell(h_S^{\phi}, z_j)] - \mathbb{E}_{\phi}[\ell(h_{S^{\setminus i}}^{\phi}, z_j)]. \qquad (33)
\end{aligned}
$$

A.6. Proof of Lemma A.4

From Lemma A.3 we have

$$
-\frac{\rho}{6}|z_1 - z_2|^3 \leq \ell(h, z_1) - \ell(h, z_2) - \langle \nabla \ell(h, z_2), z_1 - z_2 \rangle - \langle \nabla^2 \ell(h, z_2)(z_1 - z_2), z_1 - z_2 \rangle \leq \frac{\rho}{6}|z_1 - z_2|^3
$$

This gives us a lower bound on $\ell(h, z_1)$

$$
-\frac{\rho}{6}|z_1 - z_2|^3 + \ell(h, z_2) + \langle \nabla \ell(h, z_2), z_1 - z_2 \rangle + \langle \nabla^2 \ell(h, z_2)(z_1 - z_2), z_1 - z_2 \rangle \leq \ell(h, z_1) \qquad (34)
$$

Consider $z_j \in S$ such that $z_i = z_j + \alpha$ for some $j \neq i$ where $\alpha \in B_p(v)$ such that $\mathbb{E}[\alpha] = 0$ and $\mathbb{E}[\alpha^T \alpha] = 1$. Using the lower bound in Lemma A.2 with $z_1 = z_i, z_2 = z_j$ we get

$$\begin{aligned}
\mathbb{E}_\phi[\ell(h_S^\phi, z_i)] - \mathbb{E}_\phi[\ell(h_{S^{\setminus i}}^\phi, z_j)] &\leq m\beta + (4m-1)\gamma \\
&\quad + 2(m-1)\Delta \\
-\frac{\rho}{6}\|\alpha\|^3 + \mathbb{E}_\phi[\ell(h_S^\phi, z_j)] + \mathbb{E}_\phi[\langle \nabla \ell(h_S^\phi, z_j), \alpha\rangle] + \mathbb{E}_\phi[\langle \nabla^2 \ell(h_S^\phi, z_i)\alpha, \alpha\rangle] - \mathbb{E}_\phi[\ell(h_{S^{\setminus i}}^\phi, z_j)] &\leq m\beta + (4m-1)\gamma \\
&\quad + 2(m-1)\Delta \\
\mathbb{E}_\phi[\ell(h_S^\phi, z_j)] + \mathbb{E}_\phi[\langle \nabla \ell(h_S^\phi, z_j), \alpha\rangle] + \mathbb{E}_\phi[\langle \nabla^2 \ell(h_S^\phi, z_i)\alpha, \alpha\rangle] - \mathbb{E}_\phi[\ell(h_{S^{\setminus i}}^\phi, z_j)] &\leq m\beta + (4m-1)\gamma \\
&\quad + 2(m-1)\Delta + \frac{\rho}{6}\|\alpha\|^3
\end{aligned}$$

Taking Expectation over α we get

$$\begin{aligned}
\mathbb{E}_\phi[\ell(h_S^\phi, z_j)] + \mathbb{E}_{\alpha,\phi}[\langle \nabla \ell(h_S^\phi, z_j), \alpha\rangle] + \mathbb{E}_{\alpha,\phi}[\langle \nabla^2 \ell(h_S^\phi, z_i)\alpha, \alpha\rangle] - \mathbb{E}_\phi[\ell(h_{S^{\setminus i}}^\phi, z_j)] &\leq m\beta + (4m-1)\gamma \\
&\quad + 2(m-1)\Delta + \frac{\rho}{6}\|\alpha\|^3
\end{aligned}$$

Note that we can change the order of expectation due to Fubini's theorem

$$\begin{aligned}
\mathbb{E}_\phi[\ell(h_S^\phi, z_j)] + \mathbb{E}_{\phi,\alpha}[\langle \nabla^2 \ell(h_S^\phi, z_i)\alpha, \alpha\rangle] - \mathbb{E}_\phi[\ell(h_{S^{\setminus i}}^\phi, z_j)] &\leq m\beta + (4m-1)\gamma + 2(m-1)\Delta + \frac{\rho}{6}\|\alpha\|^3 \\
\mathbb{E}_\phi[\ell(h_S^\phi, z_j)] + \mathbb{E}_{\phi}[\mathrm{tr}(\nabla^2 \ell(h_S^\phi, z_i))] - \mathbb{E}_\phi[\ell(h_{S^{\setminus i}}^\phi, z_j)] &\leq m\beta + (4m-1)\gamma + 2(m-1)\Delta + \frac{\rho}{6}\|\alpha\|^3 \\
\mathbb{E}_{\phi}[\mathrm{tr}(\nabla^2 \ell(h_S^\phi, z_i))] \leq m\beta + (4m-1)\gamma + 2(m-1)\Delta + \frac{\rho}{6}\mathbb{E}[\|\alpha\|^3] &+ \mathbb{E}_\phi[\ell(h_S^\phi, z_j)] - \mathbb{E}_\phi[\ell(h_{S^{\setminus i}}^\phi, z_j)] \quad \blacksquare
\end{aligned}$$

A.7. Proof of Theorem 5.3

We start with the results of Lemma A.4. Taking Expectation over $z \sim \mathcal{D}$ we have

$$\begin{aligned}
\mathbb{E}_{z,\phi}[\mathrm{tr}(\nabla^2 \ell(h_S^\phi, z))] &\leq m\beta + (4m-1)\gamma + 2(m-1)\Delta + \frac{\rho}{6}\mathbb{E}[\|\alpha\|^3] + \mathbb{E}_{z,\phi}[\ell(h_S^\phi, z_j)] - \mathbb{E}_{z,\phi}[\ell(h_{S^{\setminus i}}^\phi, z_j)] \\
\mathbb{E}_{z,\phi}[\mathrm{tr}(\nabla^2 \ell(h_S^\phi, z))] &\leq m\beta + (4m-1)\gamma + 2(m-1)\Delta + \frac{\rho}{6}\mathbb{E}[\|\alpha\|^3] + \beta \\
\mathbb{E}_{z,\phi}[\mathrm{tr}(\nabla^2 \ell(h_S^\phi, z))] &\leq (m+1)\beta + (4m-1)\gamma + 2(m-1)\Delta + \frac{\rho}{6}\mathbb{E}[\|\alpha\|^3]
\end{aligned}$$

Using Lemma 5.2

$$\mathbb{E}_{z,\phi}[\mathrm{tr}(\nabla^2 \ell(h_S^\phi, z))] \leq L(m+1)(1-e^{-\epsilon}) + (4m-1)\gamma + 2(m-1)\Delta + \frac{\rho}{6}\mathbb{E}[\|\alpha\|^3] \quad \blacksquare$$

A.8. Privacy vs Memorization via MIA

We have considered a differential privacy setting to understand the link between memorization and privacy. However, another common approach to measure privacy is via membership inference attack (MIA), such as using Privacy Meter (Murakonda & Shokri, 2020). Membership inference attacks aim to detect if a given sample was used in training a deep neural network. While membership attacks can estimate privacy leakage, it is hard to estimate privacy guarantees. However, for completeness, we used the LiRA (Carlini et al., 2022) a recent state-of-the-art MIA to perform a black-box shadow model-based attack. LiRA trains shadow models, which are used to learn the logit-scaled probability distribution when a sample is in the training set and when not in the training set. A likelihood ratio test is performed using the learned distribution to detect whether a sample was in the train set. We detail the experiment below.

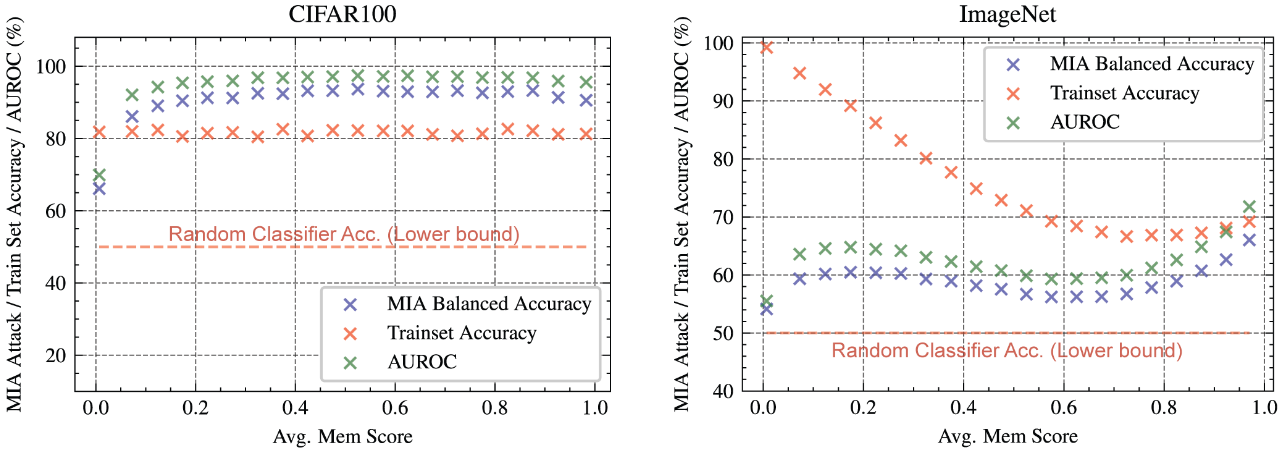


Figure 7. LiRA MIA attack with 48 shadow models on CIFAR100. Plot of accuracy of MIA attack vs memorization score of the samples.

Figure 8. LiRA MIA attack with 48 shadow models on ImageNet. Plot of accuracy of MIA attack vs memorization score of the samples.

Figure 9. Plot of MIA accuracy, AUROC and trainset accuracy vs memorization score for CIFAR100 and ImageNet datasets.

Experiment Commonly with MIA the shadow models are trained either using a subset (Nasr et al., 2019) or a similar set as the target training set. In our case, since we are interested in measuring privacy, we train the shadow models using the random sub-sampled version of the entire target train set. We use a subset ratio of 0.5 and 0.7 for CIFAR100 and ImageNet, respectively. A subset ratio of s_r denotes that each shadow model was trained on a random subset whose length was $s_r \times full\_train\_set\_length$.

We used 48 shadow models and tested 10 target (i.e. different from the 48) models trained on CIFAR100 and ImageNet to detect which samples were in the train set. For CIFAR100 we used ResNet18 models, and for ImageNet we used ResNet50 models from Feldman & Zhang (2020)'s 0.7 subset ratio. Using the 10 target models we calculated the average MIA attack success rate, AUROC, and the train set accuracy for various memorization score buckets. The grouping was similar to our previous experiments, i.e. we grouped the memorization scores and the corresponding metrics into memorization bins for visualization. This was used to plot the Figure 9. Please note an attack success rate and AUROC of 50% is random since MIA is effectively a binary classifier (in trainset or not).

Takeaways: Results in Figure 9 show that more memorized samples are easier to detect using MIA attacks. This is in line with our theoretical analysis (Theorem 5.4). This conclusion is slightly complicated for ImageNet because MIA performance also depends on the accuracy of the samples in the trainset. If the accuracy on the trainset is high and memorization is low, MIA is unsuccessful (see memorization score < 0.1 in Figure 8). If the memorization is high but the model is inaccurate (see memorization score range $0.4 - 0.7$) then due to lack of model learning MIA is unsuccessful but to a lesser extent. However, if memorization is high and accuracy on the samples is high MIA is successful (see memorization score > 0.7).

A.9. Memorization vs Other Proxies

In this paper, we explore a specific memorization proxy, that of input loss curvature. We discuss the theory for this proxy and how it relates to privacy and memorization. For completeness and to provide readers with sufficient context, we have considered three proxies previously proposed in the literature, Model Confidence (Carlini et al., 2019), Adversarial Distance (Stock & Cisse, 2018; Carlini et al., 2019) and Learning Time (Jiang et al., 2020). For better visualization, we plot (1 - normalized learning time), this lets us plot all the three metrics in a single plot for easy comparison. Normalized learning time is 1 if a sample is learnt at the last epoch, and 0 if it learns in the first epoch. We plot how each of these proxies trend with memorization scores in Figures 12 and 13.

Takeaways: For CIFAR100 all the three proxies are mostly linear with respect to memorization score, except some non-linearity at the extremes. However, this changes drastically for ImageNet, both Adversarial Distance and Model Confidence are quite non-linear, while learning time is linear. However, as far as we are aware there is no theoretical guarantees for these proxies.

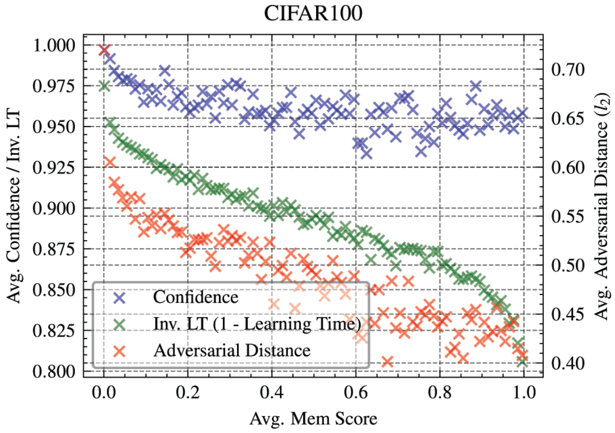


Figure 10. Comparing memorization score proxies for CIFAR100.

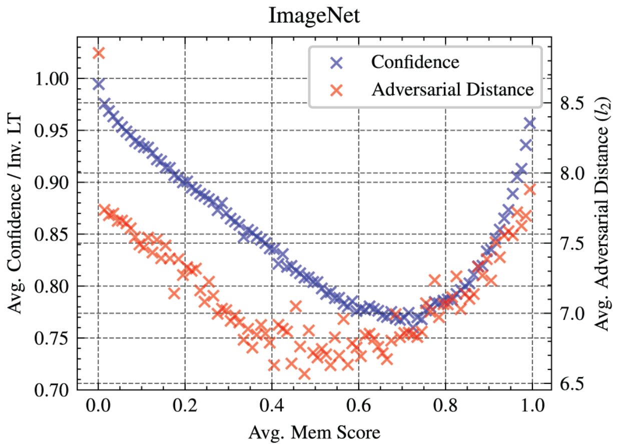


Figure 11. Comparing memorization score proxies for ImageNet.

Figure 12. For CIFAR100 all the three proxies are mostly linear with respect to memorization score. However this changes drastically for ImageNet, both Adversarial Distance and Model Confidence are quite non-linear, while learning time is linear. However, as far as we are aware there is no theoretical guarantees for these proxies.

A.10. Result of Using tr(H)

Curvature as studied in prior works (Garg & Roy, 2023; Garg et al., 2023; Moosavi-Dezfooli et al., 2019) use sum of absolute eigenvalues of H. Thus, we focus our study on the sum of absolute eigenvalues of H. However, the theoretical results do not need H to be positive semi-definite (PSD). But, assuming PSD links prior works with the theoretical analysis presented in this paper. This assumption is also quite reasonable when using CURE (Moosavi-Dezfooli et al., 2019) or adversarial training as shown by Moosavi-Dezfooli et al. (2019).

Further, assuming PSD on H for empirical curvature computations implies $tr(H) \leq tr(H^2)$ thus, we can write $Curv \propto tr(H^2)$, this approximation is similar to Garg & Roy (2023); Garg et al. (2023); Moosavi-Dezfooli et al. (2019). However, to bolster our results without needing PSD assumption we provide additional results for $Curv \propto tr(H)$ estimated using the same technique i.e. $E[v^T Hv]$ this is presented in Figures 14, 15 and 16. The results with $tr(H)$ are almost identical to those using $tr(H^2)$.

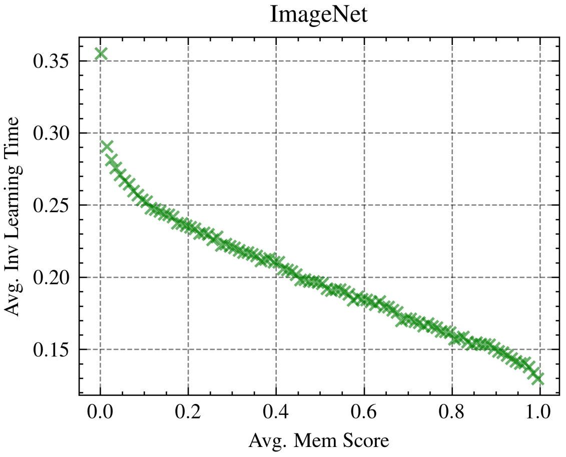


Figure 13. Plot of inverse learning time (1-LT) vs memorization score for ImageNet.

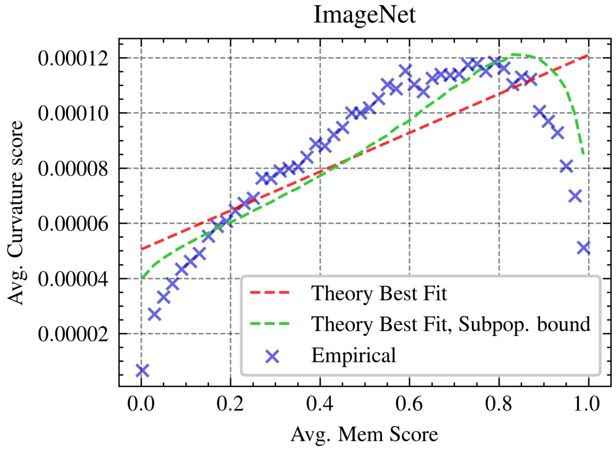


Figure 14. Plot of memorization score vs. input loss curvature at the end of training for ImageNet (average over 100 ResNet50).

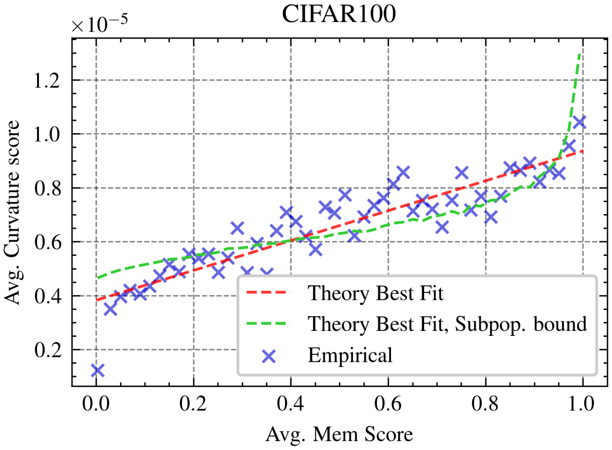


Figure 15. Plot of memorization score vs. input loss curvature at the end of training for CIFAR100 (average over 1000 Small Inception models).

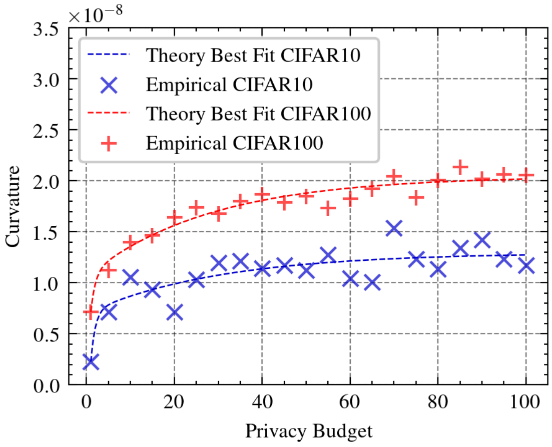


Figure 16. Plot of privacy vs. loss curvature for CIFAR10 and CIFAR100. The best-fit curve (dashed) is predicted by Theorem 5.3.