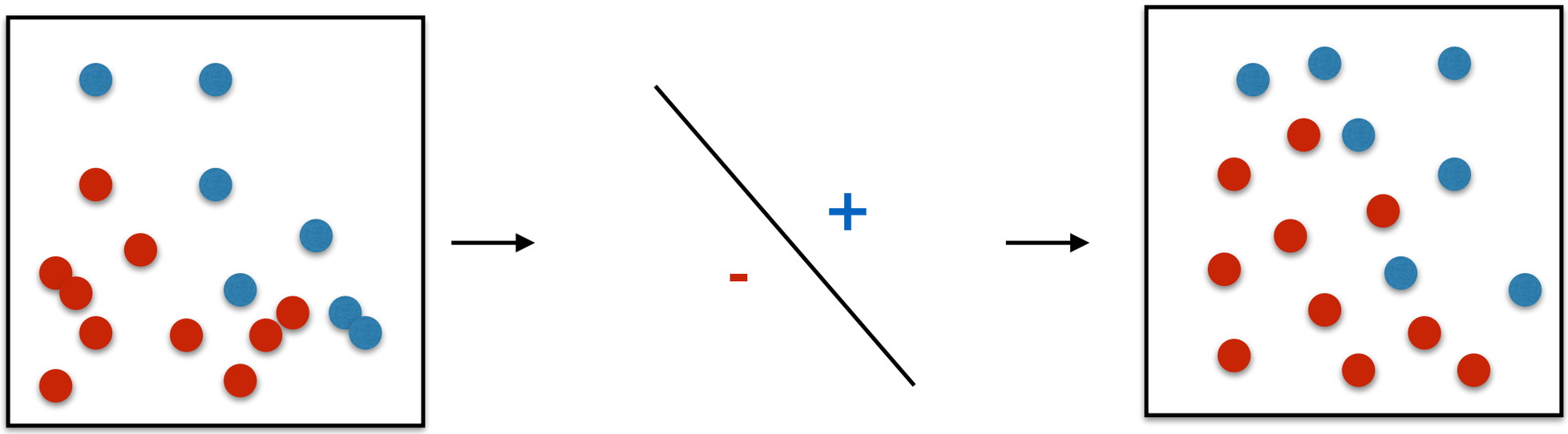
Leave-one-out Distinguishability in Machine Learning

Jiayuan Ye*, Anastasia Borovykh†, Soufiane Hayou*, Reza Shokri*

*National University of Singapore †Imperial College London

Standard Machine Learning

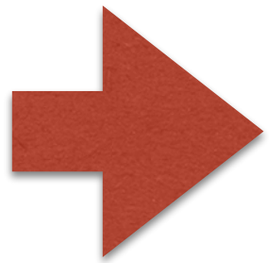


Training Data Learning a Classifier Test Data

Train and Test data Drawn i.i.d. from the same distribution
High test performance means good classifier

Issues in practice

- Robustness (influence of adversarial data)
- Privacy (information leakage)
- Sample selection bias
- Overfitting and memorization
- …



Track and control how each training data change trained model's prediction

Our work: We show they are intrinsically the *same* problem of estimating *leave-one-out distinguishability*

We propose one *analytical* framework to solve it *accurately* and *efficiently*

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 output (e.g., loss, prediction) given training dataset D and query data Q, where the randomness is over the random coins of the algorithm
- We define Leave-one-out Distinguishability as

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

where *dist* can be *any statistical distance* such as Euclidean distance between distribution means and KL divergence

Influence of Adversarial Data

- Poisoning attacks construct *poisoned* data that, when used in training with clean data, could adversely *influence* test loss
- Adversarial *training* examples further find **small** perturbation of original data that adversely ***influence*** test predictions

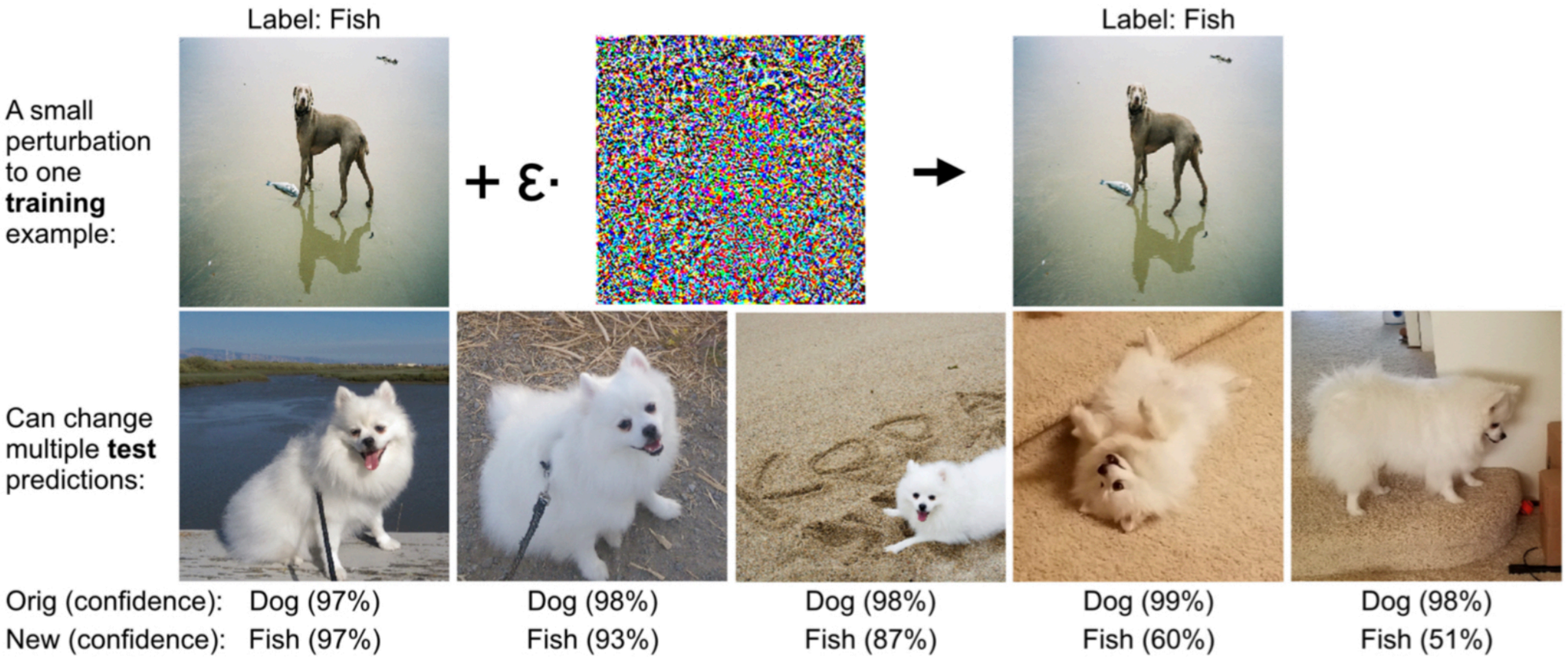


Figure 5. **Training-set attacks.** We targeted a set of 30 test images featuring the first author's dog in a variety of poses and backgrounds. By maximizing the average loss over these 30 images, we found a visually-imperceptible change to the particular training image (shown on top) that flipped predictions on 16 test images.

Koh, Pang Wei, and Percy Liang. "Understanding black-box predictions via influence functions." *International Conference on Machine Learning*. PMLR, 2017.

Influence as LOOD

- Influence of training data z on model's loss at test data z_{test} is

$$I_{loss}(z, z_{test}) = L(\hat{\theta}_{-z}; z_{test}) - L(\hat{\theta}; z_{test})$$

where $\hat{\theta}_{-z} = \arg\min_{\theta} \sum_{z_i \in D} L(z_i, \theta) - L(z, \theta)$ and $\hat{\theta} = \arg\min_{\theta} \sum_{z_i \in D} L(z_i, \theta)$

- This is leave-one-out distinguishability given by mean distance among loss distributions, for an (idealized) algorithm that outputs the optimal model given any training dataset

Computation difficulty: exact minimizer is hard to find, and computing it as is requires leave-one-out retraining

Steinhardt, Jacob, Pang Wei W. Koh, and Percy S. Liang. "Certified defenses for data poisoning attacks." *Advances in neural information processing systems* 30 (2017).
Koh, Pang Wei, and Percy Liang. "Understanding black-box predictions via influence functions." *International Conference on Machine Learning*. PMLR, 2017.

Influence Function as LOOD

- Influence function (Koh and Liang, 2017) aims to estimate this influence *without retraining* for *small perturbation* of training data, via a first-order Taylor approximation of influence

$$I_{pert,loss}(z, z_{test}) = \nabla_{\delta} L(z_{test}, \hat{\theta}_{z_\delta, -z})$$

can be computed analytically using Hessian approximations at $\hat{\theta}$

where $$\hat{\theta}_{z_\delta, -z} = \arg\min_{\theta \in \Theta} \sum_{i=1}^{n} L(z_i, \theta) - L(z, \theta) + L(z_\delta, \theta)$$

- This is partial derivative of the following mean distance LOOD

$$I_{pert,loss}(z, z_{test}) = \nabla_{\delta} LOOD_{mean}(z_{test}; D_{-z} \cup z_\delta, D)$$

Koh, Pang Wei, and Percy Liang. "Understanding black-box predictions via influence functions." *International Conference on Machine Learning*. PMLR, 2017.

Memorization intuitively quantifies "overfitting" at a per-record level

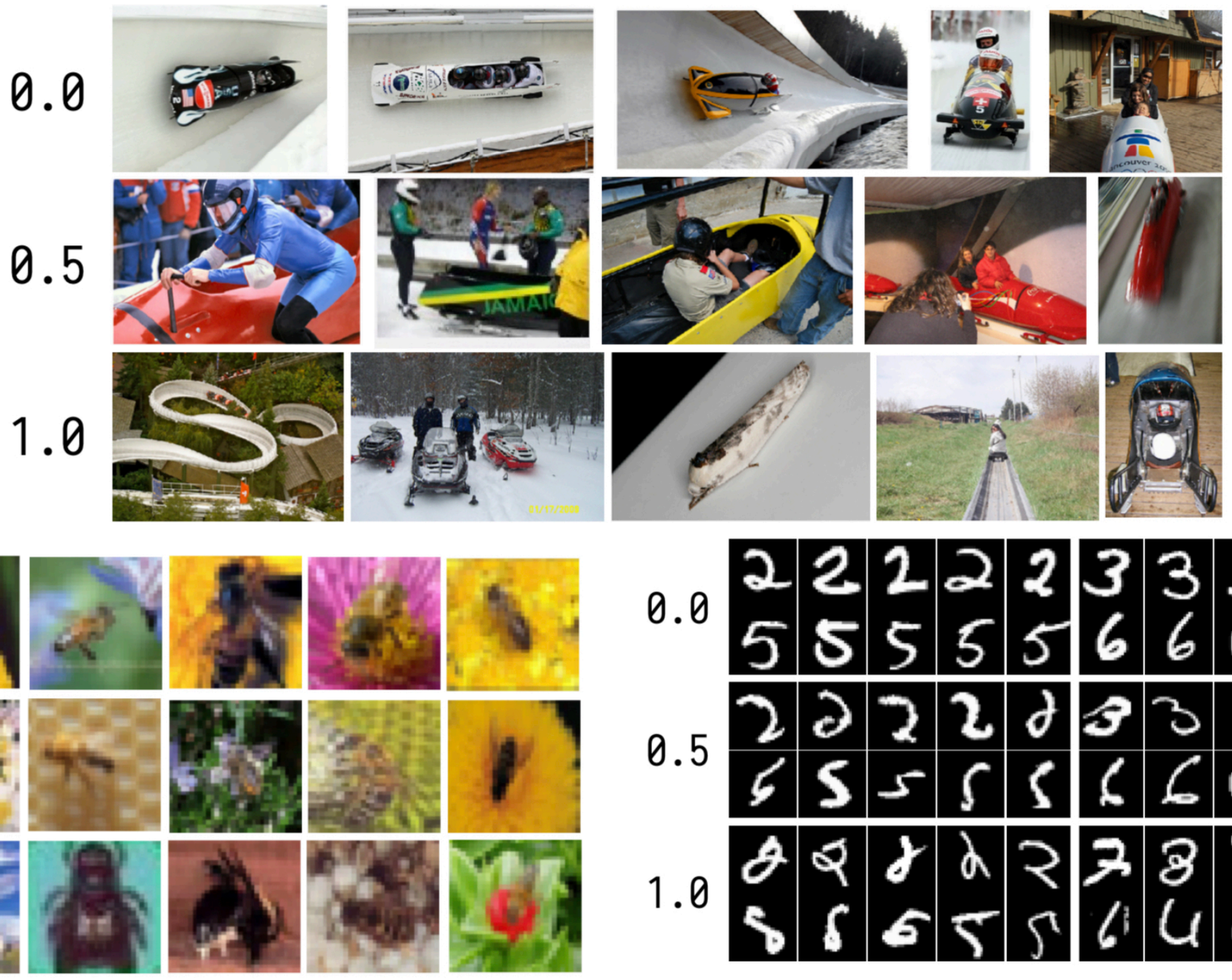


Figure 1: Examples of memorization values from ImageNet class “bobsled” (top), CIFAR-100 class “bee” (bottom left) and MNIST class 2, 3, 5, 6 (bottom right).

Feldman, Vitaly, and Chiyuan Zhang. "What Neural Networks Memorize and Why: Discovering the Long Tail via Influence Estimation." *NeurIPS*. 2020. https://pluskid.github.io/influence-memorization/

Memorization as LOOD

- Memorization is *self-influence* at $z = (x_i, y_i)$, where Influence of training data i at test data z in *prediction accuracy* is

$$\texttt{infl}(\mathcal{A}, S, i, z) := \mathop{\mathbf{Pr}}_{h \leftarrow \mathcal{A}(S)}[h(x) = y] - \mathop{\mathbf{Pr}}_{h \leftarrow \mathcal{A}(S^{\setminus i})}[h(x) = y].$$

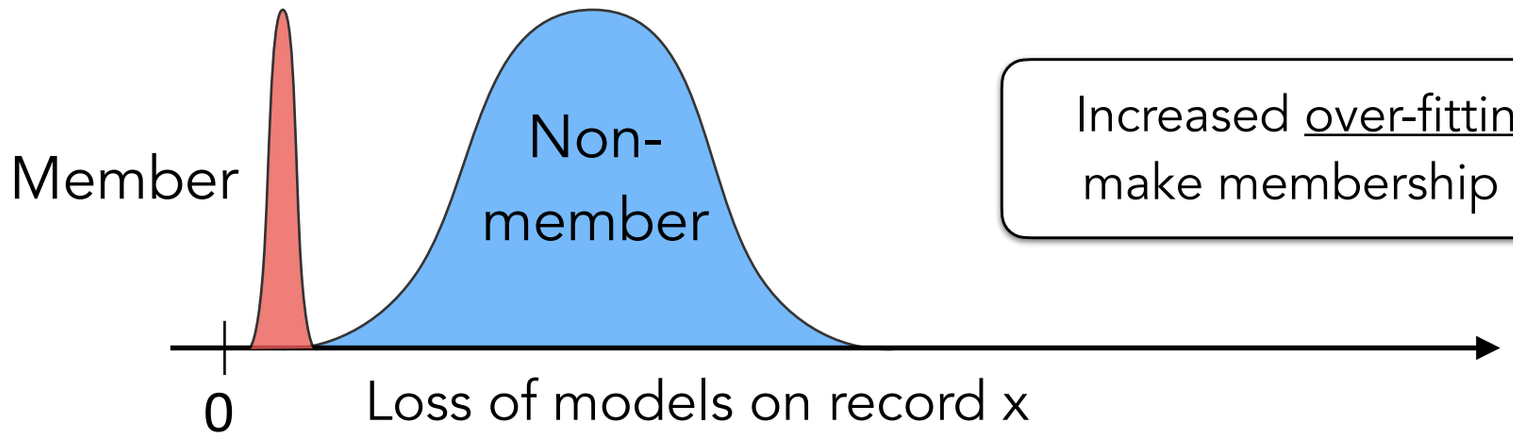
- Thus memorization is *mean distance* LOOD when query equals differing data

Computation cost: estimating the probability accurately requires multiple repeated training runs on dataset S and $S^{\setminus i}$ for each i

Feldman, Vitaly. "Does learning require memorization? a short tale about a long tail." *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. 2020.
Feldman, Vitaly, and Chiyuan Zhang. "What Neural Networks Memorize and Why: Discovering the Long Tail via Influence Estimation." *NeurIPS*. 2020.
https://pluskid.github.io/influence-memorization/

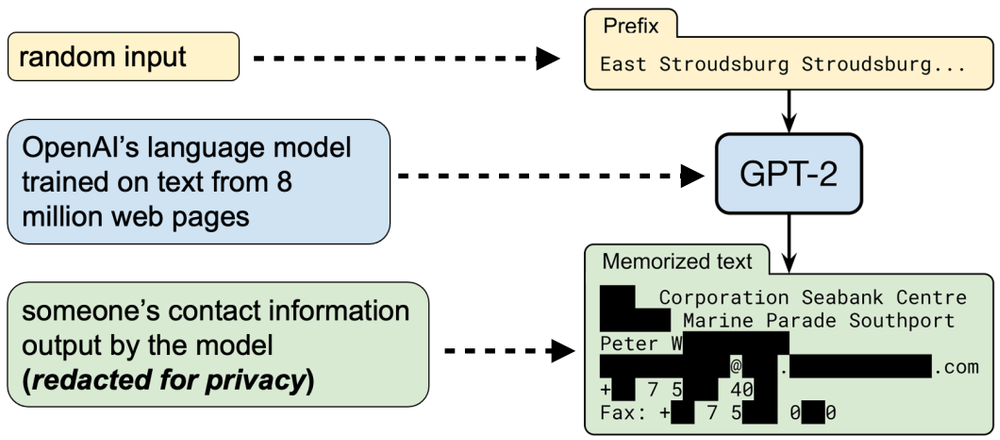
Information Leakage

- *Information Leakage* relates to the extent to which a model unintentionally reveals information about its training dataset when its prediction (distribution) at a query is observed.



Increased <u>over-fitting/memorization</u> make membership inference easier

Further enables reconstructing sensitive data used in training



[Carlini, Tramer, et al.] Extracting Training Data from Large Language Models, Usenix security'21

Information Leakage as LOOD

- Information leakage boils down to the *change* of *algorithm's output* after only *one individual* in the training dataset D changes its *participance* (denote the changed dataset by D')
- This is exactly LOOD given by divergence measures, for an algorithm that outputs model's prediction on test query Q

(ε, δ)-differential privacy of algorithm $\mathscr{A}$ $\Leftrightarrow$ $D_{f_\varepsilon}(\mathscr{A}(D)\|\mathscr{A}(D')) \leq \delta$ where $f_\varepsilon(x) = \max(0, x - e^\varepsilon)$ and $D_f(P\|Q) = \mathbb{E}_Q\left[f(\frac{dP}{dQ}\right]$

Dwork, Cynthia, et al. "Calibrating noise to sensitivity in private data analysis." *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*. Springer Berlin Heidelberg, 2006.
Balle, Borja, Gilles Barthe, and Marco Gaboardi. "Privacy amplification by subsampling: Tight analyses via couplings and divergences." *Advances in neural information processing systems* 31 (2018).

LOOD vs Leakage against Inference Attacks

- Another established method for assessing information leakage is through a *membership inference attack*
  - Attacker guess whether a specific data point S was part of the model's training dataset;
    - Leakage is quantified by the performance of the attacker, e.g., its FPR versus TPR curve, on randomly trained target models and their training/test data
- ⇒ Such leakage against MIA (and other inference attacks) can be *upper-bounded* by LOOD given by *divergence measures*

[a] Shokri, Reza, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. "Membership inference attacks against machine learning models." In IEEE S&P 2017.
[b] Balle, B., Cherubin, G., & Hayes, J. Reconstructing training data with informed adversaries. In IEEE S&P 2022.
[c] Guo, C., Sablayrolles, A., & Sanjabi, M. Analyzing privacy leakage in machine learning via multiple hypothesis testing: A lesson from fano. In ICML 2023.
[d] Yeom, S., Giacomelli, I., Fredrikson, M., & Jha, S. Privacy risk in machine learning: Analyzing the connection to overfitting. In CSF 2018.
[e] Guo, C., Karrer, B., Chaudhuri, K., & van der Maaten, L.Bounding training data reconstruction in private (deep) learning. In ICML 2022.
[f] Hayes, J., Mahloujifar, S., & Balle, B. (2023). Bounding Training Data Reconstruction in DP-SGD. In NeurIPS 2023.

Leave-one-out distinguishability lie
at the heart of estimating influence,
memorization and information leakage

How to Estimate Leakage, Influence, and Memorization

- Prior approaches are heavily based on experiments, thus suffering from
  - *approximation error* *(e.g., due to first-order Taylor expansion and high-dimensional Hessian computation)*
  - *modelling error* (e.g., suboptimal inference attacks)
  - *high computation cost* (e.g., training many models for estimating memorization)
  - *instability* (uncontrollable *experimental randomness)*

We propose *one* method that estimates LOOD *accurately* and *efficiently* — model the prediction distribution of DNNs as Gaussian process to analytically compute LOOD

Gaussian Process

- "Roughly speaking a stochastic process is a generalization of a probability distribution (which describes a finite-dimensional random variable) to functions"
- We say a function f follows a Gaussian process, if for any finite collection of inputs $x_1, \cdots, x_k$, we have that $f(x_1), \cdots, f(x_k)$ follows a multivariate Gaussian distribution
- We denote $\mu(x) = \mathbb{E}_f[f(x)]$ as the mean function, denote $K(x, x') = E_f\left[(f(x) - \mu(x)) \cdot (f(x') - \mu(x))\right]$ as the covariance function for Gaussian process $GP(\mu, K)$

Rasmussen, Carl Edward, and Christopher KI Williams. *Gaussian processes for machine learning*. Vol. 1. Cambridge, MA: MIT press, 2006.

Gaussian Process regression

- Initially, for $f \sim GP(0,K)$ the prior prediction distribution is

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right)$$

for training dataset X and test data X^*

- Given ground-truth observations $\mathbf{f}$ for training data X, by conditioning the joint prior distribution on $\mathbf{f}$, we have that

$$\mathbf{f}_* | X_*, X, \mathbf{f} \sim \mathcal{N}\big(K(X_*,X)K(X,X)^{-1}\mathbf{f},$$
$$K(X_*,X_*) - K(X_*,X)K(X,X)^{-1}K(X,X_*)\big).$$

allows analytical computation of LOOD

Rasmussen, Carl Edward, and Christopher KI Williams. *Gaussian processes for machine learning*. Vol. 1. Cambridge, MA: MIT press, 2006.

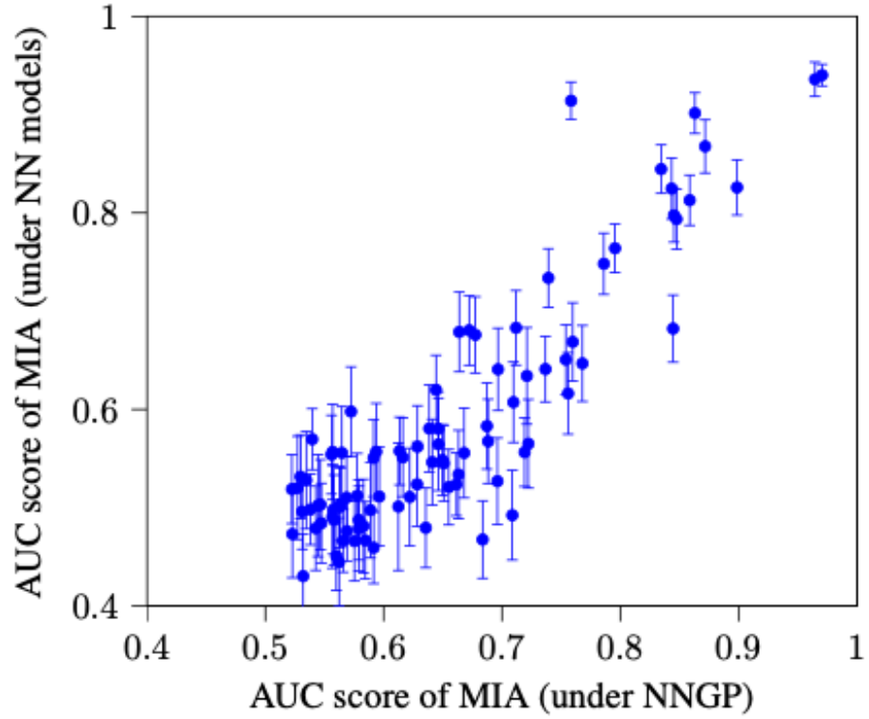
Neural Network Gaussian Processes (NNGP)

- In the limit of infinite network width, deep neural networks at random initialization forms a GP, denoted as NNGP
  - Intuitively this is due to the central limit theorem
  - The covariance function can be computed recursively over the layers, and depends on the network architecture

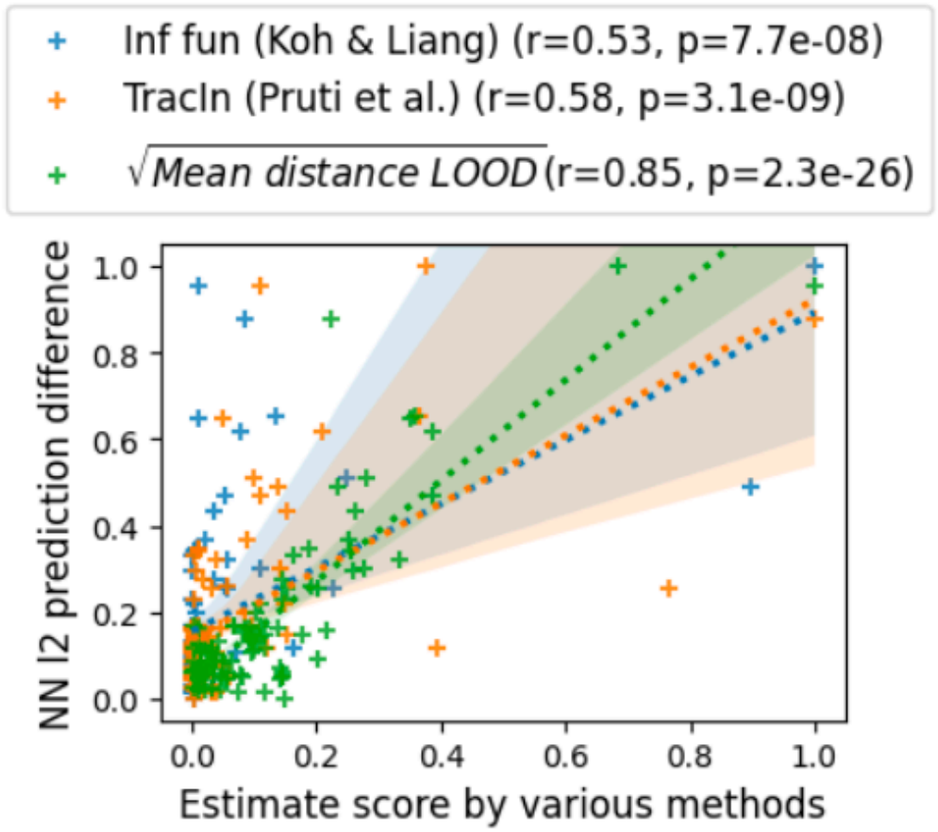
NNGP allows predictions from Bayesian neural networks to be more efficiently evaluated, and provides an analytic tool to understand *deep learning models*

Radford M. Neal. Priors for infinite networks (tech. rep. no. crg-tr-94-1). *University of Toronto*, 1994.
Christopher KI Williams. Computing with infinite networks. In *Advances in neural information processing systems*, pp. 295–301, 1997.
Lee, Jaehoon, et al. "Deep Neural Networks as Gaussian Processes." *International Conference on Learning Representations*. 2018.

LOOD under NNGP accurately estimates Leakage, Memorization and Influence



(a) MIA performance on NNGP matches NN models, for 90 random training data records



(c) Mean distance LOOD matches the prediction difference after leave-one-out retraining of NN models, for 90 randomly chosen differing data

Models are trained on 'car' and 'plane' images in the CIFAR-10 dataset

- LOOD is an analytical framework, and therefore allows efficient and accurate answers for many questions (*without* the need for training any models)
  - What prediction leaks the most information about individual training data?
  - What is the joint influence of a group of training data?
  - How does network activation choice affect leakage?
  - ...

What prediction leaks the most information?

- Optimze LOOD to identify the most influenced prediction

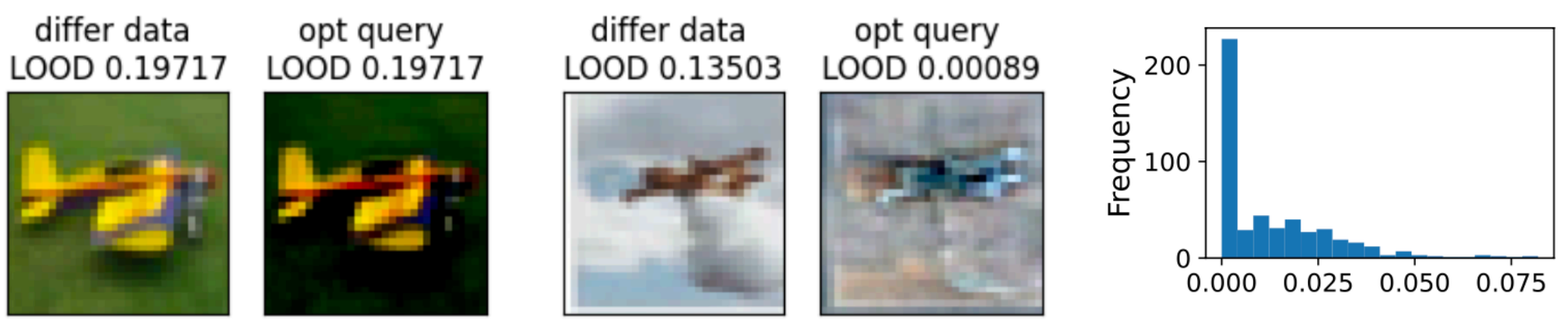
$$\arg\max_{Q} LOOD(Q; D, D') \coloneqq dist(f|_{D,\sigma^2}(Q), f|_{D',\sigma^2}(Q)),$$

- Theoretically: let S be the differing record between D and D'
  - The above optimization problem incurs stationary solution when $Q = S$ (under weak regularity conditions)
  - Mean distance LOOD is stationary at $Q = S$ only if S is very far away or very close to the training dataset D

Experimentally Find the Most-Influenced Prediction -- Reconstruction

- We can use SGD or Adam to solve the optimization problem

$$\arg\max_{Q} LOOD(Q; D, D') \coloneqq dist(f|_{D,\sigma^2}(Q), f|_{D',\sigma^2}(Q)),$$



(a) Exact reconstruction (b) Color-flipped reconstruction (c) Prediction similarity in KL (Balle et al., 2022, Section V.B.) between differ data & opt query.

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.

How the network activation choice affects leakage?

- Prior works: smooth activations enable kernels that are farther away from a low rank all-constant matrix (more expressive) than kernel obtained with non-smooth activations

- We prove that low rank kernel matrix ensures low LOOD

- Thus we prove smooth activations, e.g., GeLU, induce higher leakage (LOOD) than non-smooth activations, e.g., ReLU

Numerically Validation and Generalizability to DNNs

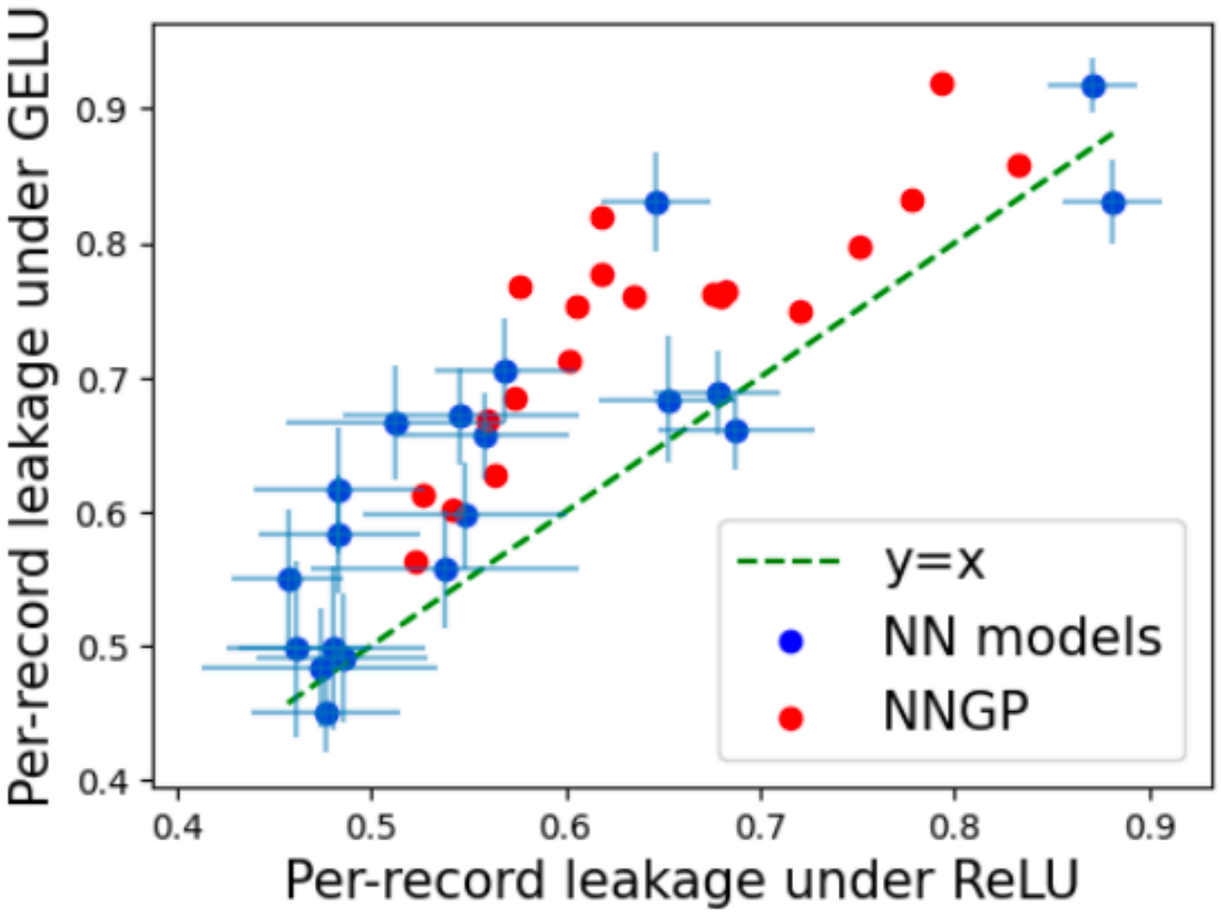


Figure 5: Empirical validation of our analytical results in of Section 5 that per-record information leakage is higher under GELU activation than under ReLU, for both NNGPs and NN models. We evaluate per-record leakage with membership inference attack performance on models trained on leave-one-out datasets.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

Conclusion

- Understanding how each training data influences the prediction of the trained model is a **canonical** problem. It allows answering
  - whether data X is used in training a model
  - how robust the model is to removing data X from its training dataset
  - how much does the model memorize individual training data
- We develop one framework -- leave-one-out distinguishability, that facilitates more *efficient* and *interpretable* answers to existing and new questions about data influence, memorization and information leakage for machine learning models