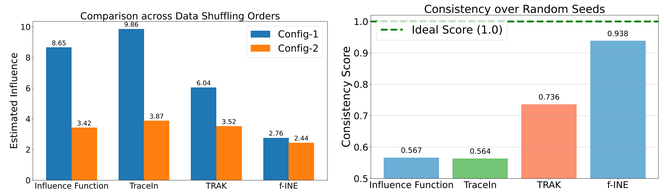
f-INE: A Hypothesis Testing Framework for Estimating Influence under Training Randomness

Subhodip Panda [1] Dhruv Tarsadiya [2] Shahswat Sourav [3] Prathosh A.P. [1] Sai Praneeth Karimireddy [2]

[1]Indian Institute of Science, India [2]University of Southern California, LA, USA [3]Washington University in St. Louis, USA

Introduction and Motivation

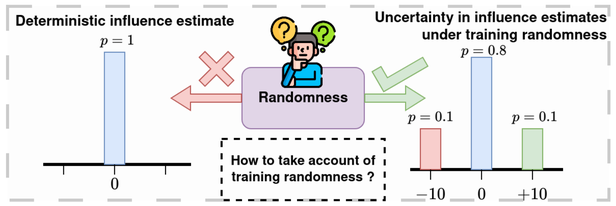
- Model predictions are shaped by the training data, making it essential to estimate how individual samples influence the outcome. Thus, influence estimation serves as an important tool for enhancing the explainability, debugging, and privacy.
- As highlighted in previous works, this decision-making tool needs to be robust to training randomness because it is used to find beneficial and harmful samples. However, current methods are not robust.



(a) Influence scores for data shuffling

(b) Consistency scores for data shuffling

- Under training, randomness influence scores are not well defined which is highlighted using the following example.



- Thus, our primary question is *How to define influence scores that are useful for decision-making even under randomness?*

Contributions

- To incorporate the training randomness, we introduce a new definition of influence termed as *f-influence*. This new definition of influence is motivated by privacy auditing and is grounded in hypothesis testing and explicitly captures training-time randomness.
- We established a bridge between influence estimation and auditing differential privacy (DP). Using this connection to DP, we prove that f-influence demonstrates useful properties such as composition and asymptotic normality.
- We then leverage these properties to design a highly scalable and efficient algorithm to estimate f-influence in a single training run using f-INfluence Estimation (f-INE) algorithm to perform poisoned data selection Llama-3.1-8B.

Problem Setup

- Let $\mathcal{D} = \{z_i\}_{i=1}^n$ denote the training dataset of n samples.
- A model parameterized by θ is optimized using a randomized algorithm (e.g., SGD) $\mathcal{A} : Z^n \to \Theta$ to achieve the trained model θ^*.
- Let $l(\theta, z_i)$ denotes the loss of the model θ on the training datum z_i.
- Our objective is to estimate the influence of a training data subset $\mathcal{S} \subseteq \mathcal{D}$ on the prediction of a test datum z_{test}.
- Let's consider the influence estimation function $\Psi_{\mathcal{A}} : \mathcal{Z} \times \mathcal{Z}^m \to \mathbb{R}$ takes a test datum z_{test}, and a subset of training data $\mathcal{S}$ to produce a score that denotes the influence of $\mathcal{S}$ on the model's prediction on z_{test}.

Hypothesis Testing Framework

- Our key insight here is that this question can be re-framed as: if I delete a suspected harmful datapoint and re-run my training, will the decrease in loss be *statistically significant* compared to what I would expect from just the training randomness?
- Thus, we define influence estimation as a binary hypothesis testing problem as follows:

$$H_0 : \mathcal{S} \text{ is influential (We train on } \mathcal{D}) \quad \text{vs.} \quad H_1 : \mathcal{S} \text{ is not influential (We train on } \mathcal{D} \setminus \mathcal{S})$$

- Essentially, we estimate the influence of $\mathcal{S}$ as the hardness of the above test.
- Now hardness of any statistical test is defined using the tradeoff curve that defines the trade-off between Type-I and Type-II errors.

Theoretical Definitions and Properties

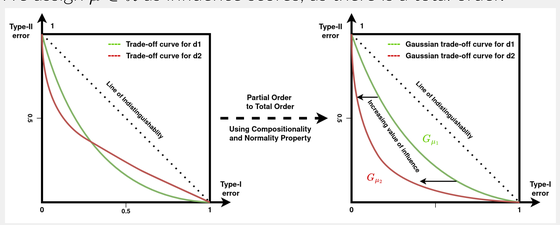
- **Definition 1:** (*f-influence*) Let f be a trade-off function. A training data subset $\mathcal{S}$ is said to be f-influential on the a test datapoint z_{test} with respect to an algorithm A, if for any test statistic $\mathcal{T}$, P denoting the distribution of $\mathcal{T}(A(\mathcal{D} \setminus \mathcal{S}), z_{test})$ and Q denoting the distribution of $\mathcal{T}(A(\mathcal{D}), z_{test})$ the following holds:

$$T(P, Q) \geq f \tag{1}$$

- **Definition 2:** (G_μ-*influence*) A training data subset $\mathcal{S}$ is G_μ-influential on the a test datapoint z_{test} with respect to an algorithm A, if for any test statistic $\mathcal{T}$, P denoting the distribution of $\mathcal{T}(A(\mathcal{D} \setminus \mathcal{S}), z_{test})$ and Q denoting the distribution of $\mathcal{T}(A(\mathcal{D}), z_{test})$ the following holds:

$$T(P, Q) \geq T(\mathcal{N}(0,1), \mathcal{N}(\mu, 1)) := G_\mu \tag{2}$$

- ***Why are these above definitions useful?***
  Ans: *f-influence* follows two important theoretical properties i.e., Compositionality and Asymptotic Normality (See the full paper for a more formal statement and proof).
- ***There is still a problem of a lack of total ordering using these definitions. Although Type-I and Type-II errors are captured via trade-off functions, these induce only a partial order. How to solve this?***
  Ans: ML training is highly iterative, and is a composition of a large number of update steps using SGD. The f-influence for any such highly composed algorithm is asymptotically always G_μ-*influence*. We assign $\mu \in \Re$ as influence scores, as there is a total order.



Full Movie Ticket!

This poster is just a trailer. For the full movie (paper), scan the following.

Scan Me

f-INE Algorithm

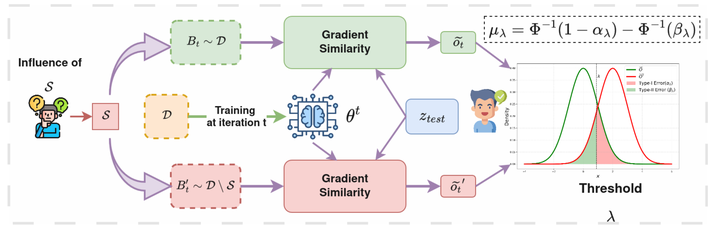
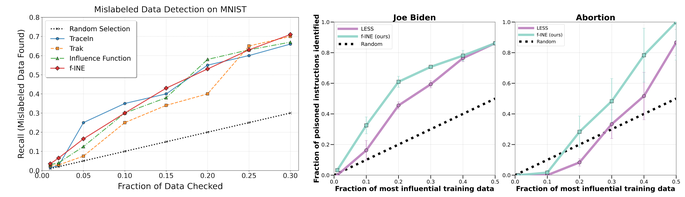
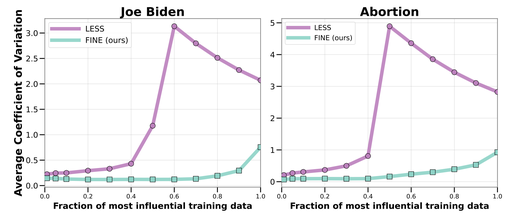


Figure 1. Overview of **f-INE** algorithm: Given a user-specified data subset $\mathcal{S}$, our method quantifies the influence of $\mathcal{S}$ as the statistical distinguishability between two distributions P and Q. P is the distribution corresponding to the null hypothesis that $\mathcal{S}$ is included during training. Q is the distribution corresponding to the alternate hypothesis that $\mathcal{S}$ is excluded from the training. In order to estimate the influence value μ, the samples from P are obtained using the model's gradient similarity of a random data-batch including $\mathcal{S}$. Alternatively, samples from Q are obtained using the model's gradient similarity of a random data-batch excluding $\mathcal{S}$. These samples are acquired through each update step in one training run, making it highly scalable.

Experimental Results

Better utility and lower variance compared to other baselines in finding mislabeled samples on the MNIST dataset and poisoned samples for the negatively instruction-tuned Llama-3.8 model on the LIMA dataset.





Conclusions and Future Works

- We reframed influence estimation as a binary hypothesis test over training-induced randomness and showed that, for composed learning procedures, the relevant object collapses to a single parameter: the Gaussian influence G_μ.
- We also combined ideas from privacy auditing with influence estimation to develop a highly scalable, efficient algorithm **f-INE**, that can estimate influence in a single training run.
- Empirically, f-INE surfaces mislabeled data on MNIST and targeted poisoned data better than baselines for Llama-3.1 LLM model on LIMA dataset, while exhibiting lower variance sensitivity to training randomness.
- Further, while our work focuses on influence estimation, the same approach can be generalized to formalize other marginal-based data valuations, such as data Shapley under training randomness.