REPHRAIN
Protecting citizens online

Differentially Private Shapley Values for Data Evaluation

Lauren Watson, University of Edinburgh

Rayna Andreeva, University of Edinburgh

Hao-Tsung Yang, University of Edinburgh

Rik Sarkar, University of Edinburgh

February 2023

UK Research and Innovation · University of Bristol · UCL · The University of Edinburgh · King's College London · University of Bath

Differentially Private Shapley Values for Data Evaluation

Lauren Watson
School of Informatics
University of Edinburgh
lauren.watson@ed.ac.uk

Rayna Andreeva
School of Informatics
University of Edinburgh
r.andreeva@sms.ed.ac.uk

Hao-Tsung Yang
School of Informatics
University of Edinburgh
haotsungyang@gmail.com

Rik Sarkar
School of Informatics
University of Edinburgh
rsarkar@inf.ed.ac.uk

Abstract

The Shapley value has been proposed as a solution to many applications in machine learning, including for equitable valuation of data. Shapley values are computationally expensive and involve the entire dataset. The query for a point's Shapley value can also compromise the statistical privacy of other data points. We observe that in machine learning problems such as empirical risk minimization, and in many learning algorithms (such as those with uniform stability), a diminishing returns property holds, where marginal benefit per data point decreases rapidly with data sample size. Based on this property, we propose a new stratified approximation method called the Layered Shapley Algorithm. We prove that this method operates on small ($O(\text{polylog}(n))$) random samples of data and small sized ($O(\log n)$) coalitions to achieve the results with guaranteed probabilistic accuracy, and can be modified to incorporate differential privacy. Experimental results show that the algorithm correctly identifies high-value data points that improve validation accuracy, and that the differentially private evaluations preserve approximate ranking of data.

1 Introduction

Large-scale machine learning and data mining depend on data contributed by various individuals and institutions. With the popularity of such data-driven systems, there is increasing awareness of the value of data and associated privacy risks. As data sharing and data marketplaces become common, it is necessary to accurately evaluate a contributor's data to provide them with the right compensation. On the other hand, identifying high-value data is also of advantage to other stakeholders such as data users. For these purposes, the Shapley value has been proposed as a fair method of determining the value of each data point [27, 37].

The Shapley value is a concept from game theory [46], defined to evaluate the contribution of individual players in a cooperative game. The concept is very general and can be applied to complex setups where multiple elements interact to produce results. Thus, it has been applied to understanding different elements in machine learning and is a popular tool in interpretable machine learning [38]. It is used to determine the importance of individual features [7, 51], neurons [22], models in ensembles [43], data points [21, 27] and many others (see [44]). However, the use of Shapley values is challenging from the perspectives of computation and privacy.

Preprint. Under review.

The computation of Shapley values requires evaluating the marginal contribution of a player (for us, a data point) with respect to all possible coalitions (subsets) of players (See Section 2). The computational problem is $\#P$-complete [13]. Additionally, each such evaluation involves training and evaluating a model. Approximations based on Monte Carlo sampling [39] of coalitions can reduce the cost to $O(n \log n)$ evaluations, which is still prohibitive in large datasets. The reduction of computation (number of evaluations) has been the focus of several recent works on the Shapley value of data [21, 27, 34].

From the privacy perspective, the Shapley value poses a complex challenge, since the contribution of a data point is influenced by the configuration of all other data in the set. Our objective is to answer a query for φ_i: Shapley value of data point i, which will require access to the entire dataset. Even Monte Carlo methods that sample random coalitions require the use of almost all data points (Section 3). Conversely, answering a query for φ_i can leak the privacy of any other data point j. To address this challenge, we develop an algorithm specifically for data evaluation that heavily samples smaller coalitions. This algorithm is more compatible with differential privacy and needs to access only a small fraction of data. Thus it is more privacy-friendly than existing methods.

Our contributions. Our approach is based on diminishing marginal gains with increasing data volume. In machine learning problems, larger training datasets are desirable, but the incremental benefit per data point decreases with increasing data size. This effect has been seen in past experiments (e.g. [49] – Fig. 4). We show theoretically, that for an empirical risk minimization (ERM) problem, the marginal reduction in loss per data point is inversely proportional to the data size i.e. $O(1/n)$ (Subsection 3.1). Thus, each new data point contributes less to the objective of loss minimization in larger datasets. Similar theoretical bounds on marginal differences hold for uniformly stable algorithms such as regularised ERM.

Using the property that the marginal utility of a data point is bounded by $O(\frac{1}{m})$ for coalitions of size m, we devise a Shapley value computation algorithm in Subsection 3.2. This algorithm stratifies the coalitions into layers and samples the layers with varying probability. We call this the *Layered Shapley Algorithm*. The algorithm heavily samples the lower layers with small coalitions, and sparsely samples the higher layers with large coalitions. The intuition is that, given the diminishing returns property, small coalitions provide sufficient information on a data point's utility. Not much remains to be gained by examining large coalitions, where the marginal utility is anyway guaranteed to be small. This algorithm relies on evaluating $O(\ln n)$ coalition samples and uses only a $O(\frac{\ln^2 n}{n})$ fraction of the dataset. It is thus highly efficient in the number of evaluations and data usage.

In Subsection 3.3 we discuss the differentially private Shapley value computation based on the Layered Shapley Algorithm. We use the strong sampling property of the algorithm with sampling-based privacy amplification results to get differential privacy at the price of relatively small noise.

Some properties of the Layered algorithm and related results are discussed in Subsection 3.4. We observe that the bias towards evaluating smaller coalitions significantly helps the computational costs since small coalitions are cheaper to train. The $O(\ln^2 n)$ data points and $O(\ln n)$ coalitions needed to compute a value φ_i, can be saved and used to answer Shapley value queries on the same dataset. Thus, the Layered algorithm produces a natural small *core set* for querying Shapley values. Finally, we argue that in the realistic case where a contributor may submit a set of points, the aggregate evaluation of the set can be carried out at a small relative cost.

Experimental results are discussed in Section 4, where we demonstrate that Shapley values calculated via the Layered Shapley Algorithm, and their differentially private counterparts, successfully describe the relative utility of data points within multiple binary classification tasks despite their reliance on small coalition sizes. The private algorithm approximately preserves the relative ranks of data points compared to the non-private version. Related works are discussed in Section 5. Proofs of theorems can be found in the appendix. In the next section, we start by reviewing the background on Shapley value and differential privacy. Readers familiar with the topics may want to quickly skim the section to note the definitions and notations.

2 Preliminaries

2.1 The Shapley value

The Shapley Value was originally developed to evaluate the contributions of different players in a cooperative game [46]. In machine learning, the set of players are usually elements of the input to the training algorithm. For example, the input features can be treated as players to estimate their relative importance. Analogously, to evaluate the relative importance of different data points, they will be treated as players and the Shapley value of a data point will represent its importance in the training process.

In a game with n players (which may be n features or n data points as the case may be), the Shapley value of player i, written as φ_i, is defined in terms of their marginal contributions to coalitions of other players. Suppose N is the set of n players, and $\mathcal{C} = 2^N$ is the set of all possible subsets (coalitions) of players. The utility obtained by any coalition C is given by a value function v, and the marginal contribution of player i with respect to c is written as $v_i(C) = v(C \cup \{i\}) - v(C)$. The Shapley value is then defined by:

$$\varphi_i(v) = \frac{1}{n} \sum_{C \subseteq N \setminus \{i\}} \binom{n-1}{|C|}^{-1} \cdot v_i(C). \tag{1}$$

Observe that this definition is equivalent to computing the average marginal gains of i over coalitions of each possible size, and then averaging over all possible sizes.

The appeal of the Shapley value is that it provides a fair allocation of credit, more meaningful than simple marginal contributions. This fairness is characterized by several intuitive properties, such as efficiency, symmetry, null player, and linearity. Shapley value is the unique valuation function that satisfies all these. See the survey [44] for details of these properties.

In the context of machine learning, v is often defined in terms of the loss function, measuring how much an element i helps in minimizing loss. In the typical data evaluation problems [21, 20, 28], each data point is treated as a player, and h is the model trained on coalition C. If $L_C(h)$ is the loss of h on C, then $v(C)$ can be defined as $v(C) = -L_C(h)$. Thus, the Shapley value φ_i is larger for data points that help more in minimizing the loss. Both the training or empirical loss [51] and validation loss [28] have been used to define v in machine learning research.

In the feature evaluation problem [25, 18], each feature is treated as a player, and for a subset C of features, $v(C)$ is defined analogously in terms of the loss.

2.1.1 Approximate computation of Shapley value

The definition of the Shapley value requires computing $v_i(C)$ for an exponential number of coalitions, making it computationally expensive. The typical approach to tractable computation is to perform a Monte Carlo estimate over the set of coalitions. Suppose π is a permutation of N, taken uniformly at random, and $\mathcal{P}_i^\pi$ is the set of items occuring before i in π. Then the basic sampling based algorithm [6] computes the average marginal gain over a sample of such subsets to obtain the approximate Shapley value: $\hat{\varphi}_i = \frac{1}{m} \sum_{j=1}^{m} v_i(\mathcal{P}_i^\pi)$. In the case of all $v_i(C)$ being bounded by a constant c, the sample complexity of $m \geq \left\lceil \frac{\ln(\frac{2}{\beta})c^2}{2\alpha^2} \right\rceil$ achieves an (α, β)-approximation guarantee [39]: $\Pr(|\hat{\varphi}_i - \varphi_i| \geq \alpha) \leq \beta$.

2.2 Differential Privacy

The privacy of data points $z \in D$ is at risk even when computing a seemingly complex aggregate value such as a machine learning model [48], or in our case a Shapley value. The computation of Shapley value φ_i uses every other data value j and thus risks their privacy. Differential privacy [14] is designed to defend against such privacy leaks. It provides a statistical privacy guarantee for all data points $z \in D$ by ensuring that the value is statistically insensitive to the presence or absence of individual data points.

Definition 2.1 (Neighbouring Databases). *Two databases D, D' are neighbouring if $H(D, D') \leq 1$, where $H(\cdot, \cdot)$ represents the hamming distance.*

Definition 2.2 (Differential Privacy [14]). *A randomized algorithm M satisfies ϵ-differential privacy if for all neighbouring databases D and D' and for all possible outputs $O \subseteq Range(M)$,* $\Pr[M(D) \in O] \leq e^{\epsilon} \cdot \Pr[M(D') \in O]$.

The sensitivity of a function f is defined to be the maximum change in the function value between neighboring databases: $\Delta f = \max_{D,D' \in \mathcal{D}} |f(D) - f(D')|$. The sensitivity determines the appropriate scale of noise to add to f to achieve differential privacy, as follows:

Theorem 2.3 (Laplace Mechanism)**.** *Given a function $f : (\mathcal{X} \times \mathcal{Y})^n \to \mathbb{R}^k$, the Laplace Mechanism releasing $f(D) + r, r \overset{k}{\sim} Lap(0, \frac{\Delta f}{\epsilon})$ satisfies ϵ-differential privacy.*

As we will discuss, one approach to releasing a privacy-preserving Shapley value is to determine its sensitivity and add the appropriate amount of noise. The challenge will be to do this while maintaining accurate estimates of the value.

Symbol	Definition	Symbol	Definition
φ_i	Shapley value of data point i	n	Data size
D, N	Data set	S	Set of permutation samples
m	Sample size	C	A coalition
$\mathcal{C}$	All coalitions	$\mathcal{C}_k$	All coalitions of size k
v	Valuation function	$v_i(C)$	Mariginal gain of i over C
α, β	Approximation parameters	ϵ	Differential privacy

Table 1: Frequently used notations.

3 Algorithms and analysis

We have discussed that the computation of Shapley values is expensive. Even with sampling-based approximations, large fractions of the dataset are used to answer a single query for φ_i. To see this, consider a single random permutation π. With probability at least $1/2$, data point i is in the second half of π. Thus with a probability of at least $1/2$, $|\mathcal{P}_i^\pi| \geq n/2$ and at least half the dataset will be required to compute a single marginal value. Since the computation of φ_i involves many such marginal valuations, nearly the entire dataset is used to answer a query for a Shapley value. In addition to the risk of exposing all data to the agent performing the computation, the large coalition sizes create challenges in terms of differential privacy, since a query for any φ_i may reveal information about any other data point.

In this section, we argue that by using the specific properties of the marginal loss in machine learning, we can improve upon these issues – with an estimation algorithm that uses only a small fraction of data. In the following subsection we analyze the intuitive idea that in larger datasets, the marginal contributions of individual data points are proportionally smaller.

3.1 Diminishing marginal gains with data

In this subsection, we discuss how increasing data volume reduces the marginal gain per data point (e.g. [49] – Fig. 4). With increasing data, the algorithm approaches the optimal model, and the loss converges to the minimum, with tinier steps. This effect can be seen more formally in the case of empirical risk minimization using the simple setup of binary classification with $0-1$ loss [45]. Given a set C of size m with labelled data points (x_i, y_i), the empirical risk of a model h is defined by $L_C(h) = \frac{1}{m} |\{i \in [m] : h(x_i) \neq y_i\}|$ – that is, the fraction of points incorrectly classified by h. The models h are drawn from a hypothesis class $\mathcal{H}$. The optimal model $h_C^\star$ in the class is the one that minimizes the loss over C. Since the loss is an average over the number of data points, the introduction of an additional data point x can only change the risk by $O(\frac{1}{m})$:

Observation 3.1. *For any subset C and new data point x, the marginal change in loss of the optimal model is bounded by* $\left| L_C(h_C^\star) - L_{C\cup\{x\}}(h_{C\cup\{x\}}^\star) \right| \leq \frac{1}{m}$.

In machine learning, a natural value function v is defined by the empirical (training) loss: $v(C) = -L_C(h_C^\star)$. Or, if an upper bound L_{max} on L is known, then possibly $v(C) = L_{max} - L_C(h_C^\star)$. In either case, for a data point i and any set C, the observation above implies a bound on the marginal gain of i w.r.t C: $v_i(C) \leq \frac{1}{m}$.

Regularized ERM and stability. The $O(1/m)$ bound on marginal difference holds more generally in stable machine learning. One of the commonly used stability notions is Uniform Stability [5]. Suppose the dataset D of size n contains points $z_i = (x_i, y_i)$ for $i \in \{1, ..., n\}$ from the domain $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. Let $D^{\setminus i}$ represent D with data point i removed. Suppose we write $\ell(A_D, z)$ to denote the loss on a point z of the model computed by algorithm A on data D. Given this, γ-Uniform Stability ensures that the change in the loss for any datapoint $z \in \mathcal{Z}$ is bounded by γ when any individual point is removed from the training set:

Definition 3.2 (Uniform stability [5])**.** *A learning algorithm A has γ-uniform stability with respect to the loss function ℓ if the following holds,*

$$\forall z \in \mathcal{Z}, \forall D \in \mathcal{Z}^m, \forall i \in \{1, ..., n\}, \|\ell(A_D, z) - \ell(A_{D^{\setminus i}}, z)\|_\infty \leq \gamma.$$

The definition implies a $O(\frac{1}{m})$ bound on the marginal gain of v. For example, $\gamma = \frac{L^2\kappa^2}{2\lambda m}$ for regularized algorithms such as L2-regularized regression in reproducing kernel Hilbert spaces with kernel $k(x, x) \leq \kappa^2$, regularization strength λ and Lipschitz constant L [5]. When v is defined to be either the averaged empirical or validation loss, this implies that the marginal difference is bounded by γ, and so is $O(\frac{1}{m})$.

Uniformly stable algorithms are known to have strong generalization properties [5, 16] and for this reason, are commonly used in research and practice. For example, regularized ERM methods such as linear and logistic regression with L2 regularization satisfy this property. Several other forms of regularizers and learning algorithms satisfy the property as well (See [5, 1]). For popular techniques such as stochastic gradient descent, there has been recent progress in establishing stability. Uniform stability with $O(\frac{1}{m})$ marginal differences is known to hold for SGD in expectation even in non-convex cases [26].

Next, we see how the property of $O(\frac{1}{m})$ marginal differences can be used to design improved algorithms for the Shapley value of data.

3.2 Layered Shapley value Algorithm

Our approach to designing an efficient algorithm is to leverage the assumption that the mariginal difference in the value of a coalition C on addition of any single data point i is bounded by $|v_i(C)| \leq \frac{c}{k}$, where k is the size of C and c is a constant independent of C.

With this assumption, we can increase the probability of small coalitions being evaluated, since the difference in value increases slowly with coalition size. The algorithm is presented as Algorithm 1. It operates by stratifying the coalitions into layers by their sizes, and then estimating the expected marginal gain from i in each layer. The algorithm is analogous to selecting m_k random coalitions from layer k. Observe that m_k drops rapidly as the coalition sizes increase. Where m_k is smaller than 1, the algorithm is probabilistically equivalent to sampling layer k with probability m_k.

Algorithm 1 Layered Shapley Algorithm

```
Input: $(\alpha, \beta)$: approximation parameters, $n$: number of points, $N$: set of points, $v$: loss function,
  $c$: constant in bound for marginal change
Output: $\hat{\varphi}_i$: the estimated Shapley value of datum $i$
for $k$ from $1$ to $n-1$ do                                          ▷ For each layer
    $m_k \leftarrow \frac{c^2}{2\alpha^2 k^2} \ln \frac{2n}{\beta}$
    $w_k \leftarrow \binom{n-1}{k}$
    $p_k \leftarrow m_k / w_k$                          ▷ Probability of a coalition in layer $k$ being used
    Draw $S_k$ where $\forall C \in \mathcal{C}_k, \Pr(C \in S_k) = p_k$     ▷ Draw a sample of coalitions from layer $k$
    $\hat{\phi}_i^k \leftarrow \frac{1}{p_k} \frac{1}{w_k} \sum_{C \in S_k} v_i(C)$              ▷ Estimate of average marginal gain in Layer $k$
end for
return $\hat{\varphi}_i = \frac{1}{n} \sum_{k=1}^{n-1} \hat{\phi}_i^k$
```

Theorem 3.3. *The estimate $\hat{\varphi}_i$ is an (α, β) approximation, that is,* $\Pr(|\hat{\varphi}_i - \varphi_i| \geq \alpha) \leq \beta$, *and is computed using a coalition sample complexity of* $O(\ln n)$.

The proof of Theorem 3.3 is in the appendix. The proof essentially relies on Hoeffding's bound to show that the estimate $\hat{\varphi}_i^k$ for each layer is within a small error, and uses the union bound to argue that the average error over all layers is probably small.

We have noted earlier that access to large data volumes for each query is undesirable. The following theorem shows that on each query, the algorithm only needs to access a small fraction of data:

Theorem 3.4. *The probability that data point j is used in the computation of φ_i is bounded by $\frac{c^2 \ln n}{2\alpha^2 n} \ln \frac{2n}{\beta}$.*

In the next section we will see how to use this result to provide differentially private Shapley values.

3.3 Differentially Private Shapley Values

Using the layered sampling approach outlined in Algorithm 1 together with the bounded marginal contributions discussed in Section 3.1, a differentially private Shapley value can be released via the Laplace mechanism [15][1]. For problems with marginal contributions bounded by $O(1/k)$, the sensitivity of the Shapley value is also bounded and can be used to ensure differential privacy. In this case, this output perturbation approach is preferable to perturbing intermediate steps of the algorithm (e.g. using private machine learning to evaluate v). This is due to both the necessity of composition over $2m$ evaluations (where m is the total number of coalitions evaluated) in that case, as it involves evaluating 2 machine learning models per sampled coalition, and the fact that private machine learning is designed in principle to mask the differences due to a single point that are measured by the marginal contribution. Instead, we combine layered sampling with the bounded sensitivity to release the private Shapley value (Algorithm 2).

Algorithm 2 Private Layered Shapley Algorithm

```
Input: $(\alpha, \beta)$: approximation parameters, $n$: number of points, $N$: set of points, $v$: loss function, $\sigma$: noise scale.
Output: $\hat{\varphi}_i^{priv}$: the private estimated Shapley value of datum $i$
$\hat{\varphi}_i$ = Layered-Shapley($(\alpha, \beta), N, v$)        ▷ Output of Algorithm 1
$\hat{\varphi}_i^{priv} = \hat{\varphi}_i + r, r \sim Lap(0, \sigma)$
return $\hat{\varphi}_i^{priv}$
```

This algorithm can be shown to be ϵ-differentially private (Thm.3.5).

Theorem 3.5. *Algorithm 2 satisfies ϵ -differential privacy with noise scale $\sigma = \frac{L^2\kappa^2}{m\lambda \ln(\frac{e^\epsilon - 1}{p} + 1)} \sum_{i=1}^n \frac{m_k}{k}$ where $p = \frac{c^2 \ln n}{2\alpha^2 n} \ln \frac{n}{\beta}$.*

The proof first demonstrates that the sensitivity of the approximated Shapley value is given by $\frac{L^2\kappa^2}{m\lambda} \sum_{i=1}^n \frac{m_k}{k}$ and then makes use of the fact that any data point has a small probability p of being used. This allows us to use results of privacy amplification by sampling [30, 4, 3] to obtain differential privacy without excessive noise.

3.4 Properties and other observations

Computation and data access costs. Compared to algorithms that use a large number of coalition samples and almost all data points, the Layered Shapley approach works with $O(\text{polylog}(n))$ data points. This is a system advantage, since accessing large datasets can incur many disk/ network/ device access costs.

Computationally, the small data requirement implies that the average coalition is only $O(\text{polylog}(n))$ in size. Since a training algorithm needs to run for each coalition, this gives a large advantage. For example, assuming that the training algorithm in question runs in ≈ $\text{poly}(n)$ time, one of the traditional approximation algorithms that require $\Omega(n)$ data points, will require a $\Omega(\text{poly}(n))$ running cost. Whereas, the Layered algorithm will run in $O(\text{polylog}(n))$ time.

[1] Note that this approach could be trivially extended to satisfy (ϵ, δ)-differential privacy via the Gaussian mechanism

Small sets for evaluations. The S_k sets generated on a run of the algorithm can be saved and treated like a core set – a small sample of a large dataset that serves to approximate results for future queries. In such a setup, each contributor needs to submit only a small fraction of their data for the general service of data evaluation. This is more privacy-friendly and likely to be acceptable for both individuals and institutions.

Valuation of data subsets. In practice, it is likely that a single contributor submits multiple data points, and the point of interest is that the total or average value (and corresponding compensation) is accurate.

If a person submits w data points, then it follows from a simple probabilistic analysis, that to ensure that the average cost is within an error of α, a sample complexity if $O(\ln(nw))$ suffices. Thus, multiple data contributions and queries for subsets effectively decrease the samples and costs.

4 Experiments

We now provide empirical results demonstrating the efficacy of our algorithms on binary classification tasks with regularized logistic regression.

Experimental Setup: Experiments were performed using both publicly available binary classification datasets and synthetic data matching the dataset used by [21].[2] The publicly available datasets used were the Adult dataset [32] and the Diabetes dataset from the UCI Machine Learning Repository [17]. The synthetic data was generated by following the synthetic data generation approach of [21] including sampling features from a 50-dimensional multidimensional Gaussian distribution $\mathcal{N}(0, I)$. All experiments use the Scikit-Learn [41] implementation of regularized logistic regression and the appropriate noise scale for $\epsilon = 1$. Private algorithm performance was reported as an average over 5 runs. In these experiments v is defined to be the negative heldout loss and coalitions with v below the random guessing baseline were not included. See the Appendix for further experimental details.

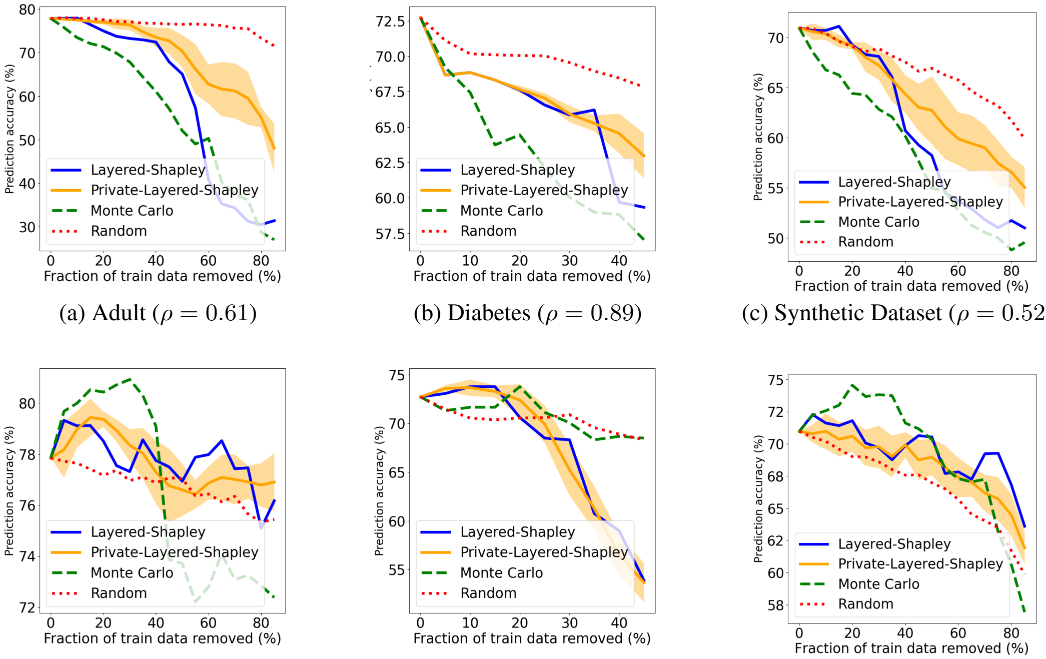
Results: As shown by Figure 1, our private and non-private Shapley value algorithms identify valuable datapoints. In the top row of Figure 1 removing datapoints with high private or non-private Shapley values results in a faster drop in classifier accuracy in comparison to removing the same number of randomly selected datapoints via the random Shapley value baseline. This implies that the identified points are of higher value to the accuracy of the classifier than randomly selected points. On the bottom row, we see that removing low value points results in a gain in accuracy initially before dropping more slowly in comparison to removing randomly selected points. For the Diabetes dataset, this holds for the bottom 20% of data only whereas for the other datasets it holds more generally. Overall, this implies that the lowest Shapley value points are those with low value for the classifier in the sense that removing them helps performance. Together, these results imply that the private Shapley values obtained contain meaningful information about the value of a given datapoint for a classifier, even in the case of differential privacy with $\epsilon = 1$.

We also report the Spearman Rank Correlation ρ between the private and non-private Layered Shapley values. The minimum value for this correlation coefficient is -1 implying a perfect negative correlation, 0 implies no correlation and 1 implies positive correlation. As the obtained values are significantly above 0, e.g. an averaged value of 0.61 or 0.89 for the Adult and Diabetes datasets respectively, this implies that the private values largely conserve the approximate rank (note that preserving rank precisely will contradict privacy).

5 Related Work

Data Valuation. Data valuation is a relatively new field and has not been widely addressed until recent years [29, 21, 20, 54, 23, 52]. One main intuition is to value the contribution of different data sources after a model is learned. The valuation can be further used to, for example, make a reasonable payment to each data contributor, which has been discussed and applied in crowdsourcing

[2] The code used in these experiments is an extension of `https://github.com/amiratag/DataShapley` [21] which is licenced under the MIT License(See `https://github.com/amiratag/DataShapley/blob/master/README.md`). The Adult dataset is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) license and the Diabetes dataset under the Creative Commons Attribution 1.0 Universal (CC0 1.0) license.



(a) Adult ($\rho = 0.61$)
(b) Diabetes ($\rho = 0.89$)
(c) Synthetic Dataset ($\rho = 0.52$)

(d) Adult ($\rho = 0.61$)
(e) Diabetes ($\rho = 0.89$)
(f) Synthetic Dataset ($\rho = 0.52$)

Figure 1: Shapley value performance demonstrated by the change in accuracy due to removing points ranked by their Shapley value (as opposed to randomly). The top row shows the performance change if the points with the highest Shapley value points are removed first. We expect meaningful Shapley values to result in lines below the red dotted random lines in this case. The bottom row shows the performance change if the smallest Shapley value points are removed first. We expect meaningful Shapley values to result in lines above the red dotted random lines in this case. The green line shows the performance of the Monte Carlo approximation of Shapley values [39]

and federated learning [29, 21, 52]. There are several algorithms for evaluating the data points such as leave-one-out testing [9], influence function estimation [31, 47, 12, 42], and core sets [11]. However, the purposes of these works are mostly for model explanation or stability improvement of models. For tasks such as rewarding the data contributors require additional properties such as fairness or privacy [36, 35]. On the other hand, Shapley value as a data valuation method provides axiomatic fairness properties [29].

Shapley value in machine learning. Shapley value has found numerous applications in machine learning [44]. Due to the hardness of the computation of exact Shapley values, approximation algorithms for Shapley values are heavily discussed. Maleki, et al. [39] provide a general bound on Shapley value with Monte Carlo sampling and show the efficiency of stratified sampling under certain assumptions. In data evaluation applications, Ghorbani, et al. proposed a framework for utilizing Shapley value in a data-sharing system [21]. Jia, et al. advanced the work with more detail and several efficient algorithms to approximate the Shapley value under different assumptions [29]. The distributional Shapley value also has been discussed in [20, 34], to address incremental updates to Shapley values, which is difficult under Monte Carlo approximation methods. Their methods calculate the Shapley value over a distribution, without revealing the true Shapley value in the output.

Several works have explored the use of Shapley values in feature selection and importance. Here, the Shapley values of features quantify how much individual features contribute to the model's performance on a set of data points [25, 18, 8, 40, 51, 50, 24, 53]. Several different approximation approaches have been proposed for the feature Shapley value. Cohen [7] assumes the number of interactions between features is significantly smaller than the combinatorial number among all features and derives the Shapley value via coalition sets with only constant sizes. Other works use the variable importance measure (VIM) to quantify the predictive value of each feature, which is called Shapley Population Variable Importance Measure (SPVIM) and can be estimated in $\Theta(d)$ time, where d is the number of features [53, 10]. In general, Shapley value has been widely used as a scoring mechanism in interpretable machine learning [44].

In comparison, our work focuses on the differentially private Shapley values of data points. To the best of our knowledge, this is the first work addressing the differential privacy of data point valuation. Shapley value of data points is a particularly challenging matter since datasets can be large and a single value computation requires many evaluations. Our algorithm operates using smaller samples of data points to obtain privacy-compatible results.

6 Conclusion

We address the privacy issue of Shapley value-based data valuation and propose the Layered Shapley value algorithm. The algorithm preserves differential privacy and utilizes the diminishing marginal gain to provide efficient computation. The theoretical bound does not extend to algorithms without uniform stability, such as training neural networks. Stability results in [26] that hold for SGD in expectation suggest that suitable results may be derived in the future. In experiments, we find that both differentially private and non-private Shapley values computed by our algorithm are still useful compared with the baseline.

We imagine a system where individuals can easily obtain valuations of their data. The theoretical results in this paper provide algorithms, but we are still far from widely usable systems. A major challenge in such a system will be to obtain meaningful value (e.g. compensation) instead of abstract numbers, which will be hard to translate to social value. In such real systems, Shapley value may or may not be the right approach. Its axiomatic properties are often cited as the reason to use it, but to what extent these hold in the Monte Carlo approximations remain to be investigated. It is also unclear if, in the case of data contributions, Shapley values agree with the human intuition of value. A few works have suggested that people may overlook the properties of the Shapley value and have incorrect expectations of it [33, 18].

Social impact. While this work contributes to the domain of ethical machine learning research by extending data valuation techniques to include privacy-preserving valuation, its social impacts can include negative elements. Both Shapley value and differential privacy are non-trivial concepts, and it is unclear if a system combining the two actually helps people, in general, make better decisions, or will simply add to greater uncertainty and fear of technology. Differential privacy, for example, does not provide absolute privacy, but rather a probabilistic one and is dependent on ϵ, which may be a source of misunderstanding. Differential privacy is also known to have disparate impact [2, 19], and it is unclear if in this case, the algorithm will maintain the fairness of valuations.

The use of such data valuation services themselves may be susceptible to attacks, frauds, and abuse. Unethical players may contribute spurious data points with the objective of increasing their own values or disrupting that of others. Valuation systems may perpetuate fraud and introduce the issue of monitoring the agent performing valuations. Leaked data valuations may make high-value data holders subject to attacks and fraud. The idea of incentivizing the contribution of (high value) data, while useful in theory, comes with some potential for abuse. Depending on the circumstances, it can be seen as a coercion to contribute data or a penalty for not contributing data. Specifically, high-value data is also likely to be privacy sensitive, and the incentives can be seen as a push toward loss of privacy.

References

[1] J. Audiffren and H. Kadri. Stability of multi-task kernel regression algorithms. In *Asian Conference on Machine Learning*, pages 1–16. PMLR, 2013.

[2] E. Bagdasaryan, O. Poursaeed, and V. Shmatikov. Differential privacy has disparate impact on model accuracy. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] B. Balle, G. Barthe, and M. Gaboardi. Privacy amplification by subsampling: tight analyses via couplings and divergences. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6280–6290, 2018.

[4] A. Beimel, H. Brenner, S. P. Kasiviswanathan, and K. Nissim. Bounds on the sample complexity for private learning and private data release. *Machine learning*, 94(3):401–437, 2014.

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

[6] J. Castro, D. Gómez, and J. Tejada. Polynomial calculation of the shapley value based on sampling. *Computers & Operations Research*, 36(5):1726–1730, 2009.

[7] S. Cohen, G. Dror, and E. Ruppin. Feature selection via coalitional game theory. *Neural Computation*, 19(7):1939–1961, 2007.

[8] S. Cohen, E. Ruppin, and G. Dror. Feature Selection Based on the Shapley Value. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, page 665–670, 2005.

[9] R. D. Cook. Detection of influential observation in linear regression. *Technometrics*, 1977.

[10] I. Covert and S.-I. Lee. Improving KernelSHAP: Practical Shapley Value Estimation Using Linear Regression. In *International Conference on Artificial Intelligence and Statistics*, pages 3457–3465, 2021.

[11] A. Dasgupta, P. Drineas, , et al. Sampling algorithms and coresets for \ell_p regression. *SIAM Journal on Computing*, pages 2060–2078, 2009.

[12] A. Datta, A. Datta, et al. Influence in Classification via Cooperative Game Theory. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[13] X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of operations research*, 19(2):257–266, 1994.

[14] C. Dwork. Differential privacy. In *Automata, Languages and Programming*, pages 1–12. ICALP, 2006.

[15] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[16] V. Feldman and J. Vondrak. Generalization bounds for uniformly stable algorithms. *Advances in Neural Information Processing Systems*, 31, 2018.

[17] A. Frank and A. Asuncion. Uci machine learning repository [http://archive. ics. uci. edu/ml]. irvine, ca: University of california. *School of information and computer science*, 213(2), 2010.

[18] D. Fryer, I. Strümke, and H. Nguyen. Shapley Values for Feature Selection: the Good, the Bad, and the Axioms. *arXiv:2102.10936*, 2021.

[19] G. Ganev, B. Oprisanu, and E. De Cristofaro. Robin hood and matthew effects–differential privacy has disparate impact on synthetic data. *arXiv preprint arXiv:2109.11429*, 2021.

[20] A. Ghorbani, M. Kim, and J. Zou. A distributional framework for data valuation. In *International Conference on Machine Learning*, pages 3535–3544. PMLR, 2020.

[21] A. Ghorbani and J. Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251. PMLR, 2019.

[22] A. Ghorbani and J. Zou. Neuron Shapley: Discovering the Responsible Neurons. In *Advances in Neural Information Processing Systems*, pages 5922–5932, 2020.

[23] J. González Cabañas, Á. Cuevas, and R. Cuevas. Fdvt: Data valuation tool for facebook users. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3799–3809, 2017.

[24] R. Guha, A. H. Khan, et al. Cga: A new feature selection model for visual human action recognition. *Neural Computing and Applications*, 33(10):5267–5286, 2021.

[25] I. Guyon and A. Elisseeff. An Introduction to Variable and Feature Selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

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

[27] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. M. Gurel, B. Li, C. Zhang, C. J. Spanos, and D. Song. Efficient task-specific data valuation for nearest neighbor algorithms. *arXiv preprint arXiv:1908.08619*, 2019.

[28] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. Hynes, N. M. Gürel, B. Li, C. Zhang, D. Song, and C. J. Spanos. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1167–1176. PMLR, 2019.

[29] R. Jia, X. Sun, J. Xu, C. Zhang, B. Li, and D. Song. An empirical and comparative analysis of data valuation with scalable algorithms. 2019.

[30] G. Kellaris and S. Papadopoulos. Practical differential privacy via grouping and smoothing. *Proceedings of the VLDB Endowment*, 6(5):301–312, 2013.

[31] P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894, 2017.

[32] R. Kohavi et al. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, pages 202–207, 1996.

[33] I. E. Kumar, S. Venkatasubramanian, C. Scheidegger, and S. Friedler. Problems with shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning*, pages 5491–5500. PMLR, 2020.

[34] Y. Kwon, M. A. Rivas, and J. Zou. Efficient computation and analysis of distributional shapley values. In *International Conference on Artificial Intelligence and Statistics*, pages 793–801. PMLR, 2021.

[35] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, et al. A survey on federated learning systems: vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[36] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, pages 50–60, 2020.

[37] Z. Liu, Y. Chen, H. Yu, Y. Liu, and L. Cui. GTG-Shapley: Efficient and Accurate Participant Contribution Evaluation in Federated Learning. *arXiv:2109.02053*, 2021.

[38] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pages 4768–4777, 2017.

[39] S. Maleki, L. Tran-Thanh, G. Hines, T. Rahwan, and A. Rogers. Bounding the Estimation Error of Sampling-based Shapley Value Approximation. *arXiv:1306.4265*, 2013.

[40] R. Patel, M. Garnelo, I. Gemp, et al. Game-Theoretic Vocabulary Selection via the Shapley Value and Banzhaf Index. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 2789–2798, 2021.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[42] G. Pruthi, F. Liu, S. Kale, and M. Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33:19920–19930, 2020.

[43] B. Rozemberczki and R. Sarkar. The Shapley Value of Classifiers in Ensemble Games. In *Proceedings of the 30th International Conference on Information and Knowledge Management*, page 1558–1567, 2021.

[44] B. Rozemberczki, L. Watson, P. Bayer, H.-T. Yang, O. Kiss, S. Nilsson, and R. Sarkar. The shapley value in machine learning. *arXiv preprint arXiv:2202.05594*, 2022.

[45] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[46] L. Shapley. A Value for N-Person Games. *Contributions to the Theory of Games*, pages 307–317, 1953.

[47] B. Sharchilev, Y. Ustinovskiy, et al. Finding influential training samples for gradient boosted decision trees. In *International Conference on Machine Learning*, pages 4577–4585, 2018.

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

[49] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[50] X. Sun, Y. Liu, J. Li, et al. Feature Evaluation and Selection with Cooperative Game Theory. *Pattern recognition*, 45(8):2992–3002, 2012.

[51] S. Tripathi, N. Hemachandra, and P. Trivedi. Interpretable Feature Subset Selection: A Shapley Value Based Approach. In *IEEE International Conference on Big Data*, pages 5463–5472, 2020.

[52] S. Wei, Y. Tong, Z. Zhou, and T. Song. Efficient and fair data valuation for horizontal federated learning. In *Federated Learning*, pages 139–152. Springer, 2020.

[53] B. Williamson and J. Feng. Efficient Nonparametric Statistical Inference on Population Feature Importance Using Shapley Values. In *International Conference on Machine Learning*, pages 10282–10291, 2020.

[54] J. Yoon, S. Arik, and T. Pfister. Data valuation using reinforcement learning. In *International Conference on Machine Learning*, pages 10842–10851. PMLR, 2020.

A Proof of Observation 3.1

First observe that for any h, $\left|L_C(h) - L_{C\cup\{x\}}(h)\right| \leq \frac{1}{m}$. This is because, if x is correctly classified by h then the difference of the two losses is $\frac{w}{m} - \frac{w}{m+1}$ where w is the number of incorrect classifications. Since $w \leq m$, the difference is at most $1/m$. On the other hand, if x is classified incorrectly, then the difference is $\frac{w}{m} - \frac{w+1}{m+1} \leq 1/m$.

In the case when with introduction of x the optimal hypothesis does not change, that is $h^\star_{C\cup\{x\}} = h^\star_C$, the observation above applies directly and the difference is at most $\frac{1}{m}$.

Now, in the event when $h^\star_{C\cup\{x\}} \neq h^\star_C$, we consider two cases. Case 1, is if $L_C(h^\star_C) \leq L_{C\cup\{x\}}(h^\star_{C\cup\{x\}})$. We know that $L_{C\cup\{x\}}(h^\star_{C\cup\{x\}}) \leq L_{C\cup\{x\}}(h^\star_C)$ and that $L_{C\cup\{x\}}(h^\star_C) - L_C(h^\star_C) \leq \frac{1}{m}$. Therefore $L_{C\cup\{x\}}(h^\star_{C\cup\{x\}}) - L_C(h^\star_C) \leq \frac{1}{m}$. Similarly, in case 2: $L_{C\cup\{x\}}(h^\star_{C\cup\{x\}}) \leq L_C(h^\star_C)$, we know that $L_C(h^\star_C) \leq L_C(h^\star_{C\cup\{x\}})$. Which implies that $L_C(h^\star_C) - L_{C\cup\{x\}}(h^\star_{C\cup\{x\}}) \leq \frac{1}{m}$.

B Proof of Theorem 3.2

In a sample from layer k, the probability that a coalition containing j is used is: $\frac{\binom{n-1}{k-1}}{\binom{n}{k}} = \frac{k}{n}$.

Thus, given $m_k = \frac{c^2}{2\alpha^2 k^2} \ln \frac{2n}{\beta}$ expected number of samples at layer k (see discussion above), the probability that j appears in a sampled coalition from stratum k is $P(j|k) \leq \frac{c^2}{2\alpha^2 kn} \ln \frac{2n}{\beta}$. Thus, over the all n strata, the probability

$$\begin{aligned} P(j) &\leq \sum_{k=1}^{n} \frac{c^2}{2\alpha^2 kn} \ln \frac{2n}{\beta} \\ &= \frac{c^2}{2\alpha^2 n} \ln \frac{2n}{\beta} \sum_{k=1}^{n} \frac{1}{k} \\ &= \frac{c^2 \ln n}{2\alpha^2 n} \ln \frac{2n}{\beta} \end{aligned}$$

C Proof of Theorem 3.3.

We first show the correctness of $(\alpha, \beta)-$approximation.

Lemma C.1. *The estimate of shapley value $\hat{\varphi}_i$ is an α, β approximation of the true shapley value φ_i. That is,* $\Pr(|\hat{\varphi}_i - \varphi_i| \geq \alpha) \leq \beta$*.*

Proof. Consider the estimate $\hat{\phi}_i^k$ at any layer k. By Algorithm 1, $\hat{\phi}_i^k = \frac{1}{m_k} \sum_{C\in S_k} v_i(C)$.

Since $E[\hat{\phi}_i^k] = \phi_i^k$, using the Chernoff-Hoeffding bound, we have $\Pr(\left|\hat{\phi}_i^k - \phi_i^k\right| \geq \alpha) \leq \Pr(\left|m_k\hat{\phi}_i^k - m_k\phi_i^k\right| \geq m_k\alpha) \leq 2\exp\left(-\frac{2\alpha^2 m_k^2}{m_k \frac{c^2}{k^2}}\right)$. Substituting the expression for m_k, we get that $\Pr(\left|\hat{\phi}_i^k - \phi_i^k\right| \geq \alpha) \leq \frac{\beta}{n}$.

By union bound, the probability that $\exists k : \Pr(\left|\hat{\phi}_i^k - \phi_i^k\right| \geq \alpha) \leq \sum_{k=1}^n \Pr(\left|\hat{\phi}_i^k - \phi_i^k\right| \geq \alpha) \leq \beta$

Observe the event $\sum_{k=1}^n \left|\hat{\phi}_i^k - \phi_i^k\right| \geq n\alpha$ requires that $\exists k : \Pr(\left|\hat{\phi}_i^k - \phi_i^k\right| \geq \alpha)$. Thus:

$$\Pr(\sum_{k=1}^n \left|\hat{\phi}_i^k - \phi_i^k\right| \geq n\alpha) \leq \beta$$
$$\implies \Pr(\frac{1}{n}\sum_{k=1}^n \left|\hat{\phi}_i^k - \phi_i^k\right| \geq \alpha) \leq \beta$$

Now, we can rewrite $\hat{\varphi}_i - \varphi_i$ as $\frac{1}{n}\sum_{k=1}^n(\hat{\phi}_i^k - \phi_i^k)$. Thus:

$$|\hat{\varphi}_i - \varphi_i| = \left|\frac{1}{n}\sum_{k=1}^n(\hat{\phi}_i^k - \phi_i^k)\right| \leq \frac{1}{n}\sum_{k=1}^n \left|(\hat{\phi}_i^k - \phi_i^k)\right|$$
$$\implies \Pr(|\hat{\varphi}_i - \varphi_i| \geq \alpha) \leq \Pr(\frac{1}{n}\sum_{k=1}^n \left|(\hat{\phi}_i^k - \phi_i^k)\right|) \leq \beta.$$

□

Now observe that since each of w_k items in layer k is sampled with probability p_k, the expected number of samples in layer k is m_k. The sample complexity follows from the summation of the sample complexity of the n layers. That is, the sample complexity $m = \sum_{k=1}^n m_k = \sum_{k=1}^n \frac{c^2}{2\alpha^2 k^2} \ln\frac{2n}{\beta} \leq \frac{c^2}{2\alpha^2}\ln\frac{2n}{\beta} \cdot \frac{\pi^2}{6}$. Combining with Lemma C.1 gives us the theorem.

D Proof of Theorem 3.5

Denote the set of all sampled coalitions of size k in Algorithm 1 by $\mathcal{C}_k$ and let $m_k = |\mathcal{C}_k|$. Assume that that v is the loss function $\ell(\cdot)$ and that in a given marginal contribution evaluation, the learning algorithm uses a sampled coalition $j \in \mathcal{C}_k$ of size k as its training set. Denote the uniform stability of the learning algorithm using a training set with datasize k by γ_k. Suppose D' is a neighbouring dataset of D, differing in at most a single point that is not the point x_i being evaluated.

When coalition sample j has k data points, let us denote the set of points in it by $D_{(k,j)} \subseteq D$. Any neighbor of it is written as $D'_{(k,j)} \subseteq D'$ respectively. $D'_{(k,j)}$ and $D_{(k,j)}$ can differ in at most a single datapoint and so are also neighbouring datasets.

Due to the Laplace Mechanism [15], noise of scale $\frac{\Delta\hat{\varphi}_i}{\epsilon}$ will suffice to guarantee ϵ-differential privacy, where $\Delta\hat{\varphi}_i$ is the sensitivity of the estimated Shapley Value. By stating $\hat{\varphi}_i$ in terms of the marginal contributions, we can bound the sensitivity as follows,

$$
\begin{aligned}
\Delta\hat{\varphi}_i = \quad & \max_{D,D'}\Big\| \frac{1}{n}\sum_{k=1}^{n-1}\frac{1}{m_k}\sum_{j\in\mathcal{C}_k}\big(v(D_{(k,j)}\cup x_i) - v(D_{(k,j)})\big) \\
& -\frac{1}{n}\sum_{k=1}^{n-1}\frac{1}{m_k}\sum_{j\in\mathcal{C}_k}\Big(v(D'_{(k,j)}\cup x_i) - v(D'_{(k,j)})\Big)\Big\| \\
= \quad & \max_{D,D'}\Big\| \frac{1}{n}\sum_{k=1}^{n-1}\frac{1}{m_k}\sum_{j\in\mathcal{C}_k}\Big(v(D_{(k,j)}\cup x_i) - v(D'_{(k,j)}\cup x_i)\Big) \\
& +\frac{1}{n}\sum_{k=1}^{n-1}\frac{1}{m_k}\sum_{j\in\mathcal{C}_k}\Big(v(D'_{(k,j)}) - v(D_{(k,j)})\Big)\Big\| \\
\leq \quad & \max_{D,D'}\Big\| \frac{1}{n}\sum_{k=1}^{n-1}\frac{1}{m_k}\sum_{j\in\mathcal{C}_k}\Big(v(D_{(k,j)}\cup x_i) - v(D'_{(k,j)}\cup x_i)\Big)\Big\| \\
& +\max_{D,D'}\Big\| \frac{1}{n}\sum_{k=1}^{n-1}\frac{1}{m_k}\sum_{j\in\mathcal{C}_k}\Big(v(D'_{(k,j)}) - v(D_{(k,j)})\Big)\Big\| \\
\leq \quad & \frac{1}{n}\sum_{k=1}^{n-1}\frac{1}{m_k}\sum_{j\in\mathcal{C}_k}\gamma_{k+1} + \frac{1}{n}\sum_{k=1}^{n-1}\frac{1}{m_k}\sum_{j\in\mathcal{C}_k}\gamma_k \\
\leq \quad & \frac{2}{n}\sum_{k=1}^{n-1}\frac{1}{m_k}\sum_{j\in C_k}\gamma_k \\
= \quad & \frac{2}{n}\sum_{k=1}^{n-1}\gamma_k \\
= \quad & \frac{L^2\kappa^2}{n\lambda}\sum_{k=1}^{n-1}\frac{1}{k}
\end{aligned}
$$

The last three inequalities follow from the fact that the sensitivity is $\propto \frac{1}{k}$ for a coalition of size k and use the uniform stability bound for regularized algorithms given by [5].

Finally, due to Theorem 3.4 and amplification by sampling [30, 4, 3], Laplace noise with scale $\frac{L^2\kappa^2}{n\lambda\ln(\frac{e^\epsilon-1}{p}+1)}\sum_{k=1}^{n-1}\frac{1}{k}$ suffices with $p = \frac{c^2\ln n}{2\alpha^2 n}\ln\frac{2n}{\beta}$.

Note that this sensitivity and noise scale are asymptotically better than what was stated in the theorem statement, since in our algorithm, $m \in O(\mathrm{polylog}(n))$. The main body of the paper will be updated in the camera ready version.

E Further Experimental Details

Parameters: Experiments use the following settings: $\epsilon = 1$, $\alpha = 0.05$, $\beta = 0.05$, $\lambda = 1.0$ and $|D| = 100$. The data is normalized to the range $(0, 1)$ in order to bound κ.

Compute: The compute requirements of these experiments were low, all experiments were run on laptops using 2.9 GHz Quad-Core Intel Core i7 processors.