Data Banzhaf: A Robust Data Valuation Framework for Machine Learning

Jiachen T. Wang
Princeton University

Ruoxi Jia
Virginia Tech

Abstract

Data valuation has wide use cases in machine learning, including improving data quality and creating economic incentives for data sharing. This paper studies the robustness of data valuation to noisy model performance scores. Particularly, we find that the inherent randomness of the widely used stochastic gradient descent can cause existing data value notions (e.g., the Shapley value and the Leave-one-out error) to produce inconsistent data value rankings across different runs. To address this challenge, we introduce the concept of safety margin, which measures the robustness of a data value notion. We show that the Banzhaf value, a famous value notion that originated from cooperative game theory literature, achieves the largest safety margin among all semivalues (a class of value notions that satisfy crucial properties entailed by ML applications and include the famous Shapley value and Leave-one-out error). We propose an algorithm to efficiently estimate the Banzhaf value based on the Maximum Sample Reuse (MSR) principle. Our evaluation demonstrates that the Banzhaf value outperforms the existing semivalue-based data value notions on several ML tasks such as learning with weighted samples and noisy label detection. Overall, our study suggests that when the underlying ML algorithm is stochastic, the Banzhaf value is a promising alternative to the other semivalue-based data value schemes given its computational advantage and ability to robustly differentiate data quality.

1 INTRODUCTION

Data valuation, i.e., quantifying the usefulness of a data source, is an essential component in developing machine learning (ML) applications. For instance, evaluating the worth of data plays a vital role in cleaning bad data (Tang et al., 2021; Karlaš et al., 2022) and understanding the model's test-time behavior (Koh and Liang, 2017). Furthermore, determining the value of data is crucial in creating incentives for data sharing and in implementing policies regarding the monetization of personal data (Ghorbani and Zou, 2019; Zhu et al., 2019).

Due to the great potential in real applications, there has been a surge of research efforts on developing data value notions for supervised ML (Jia et al., 2019b; Ghorbani and Zou, 2019; Yan and Procaccia, 2020; Ghorbani et al., 2021; Kwon and Zou, 2021; Yoon et al., 2020). In the ML context, a data point's value depends on other data points used in model training. For instance, a data point's value will decrease if we add extra data points that are similar to the existing one into the training set. To accommodate this interplay, current data valuation techniques typically start by defining the "utility" of a *set* of data points, and then measure the value of an *individual* data point based on the change of utility when the point is added to an existing dataset. For ML tasks, the utility of a dataset is naturally chosen to be the performance score (e.g., test accuracy) of a model trained on the dataset.

However, the utility scores can be noisy and unreliable. Stochastic training methods such as stochastic gradient descent (SGD) are widely adopted in ML, especially for deep learning. The models trained with stochastic methods are inherently random, and so are their performance scores. This, in turn, makes the data values calculated from the performance scores *noisy*. Despite being ignored in past research, we find that the noise in a typical learning process is actually substantial enough to make different runs of the same data valuation algorithm produce inconsistent value rankings. Such inconsistency can pose challenges for building reliable applications based on the data value scores and rankings, e.g., low-quality data identification.

In this paper, we initiate the study of the robustness of data valuation to noisy model performance scores. Our technical contributions are listed as follows.

Theoretical framework for quantifying robustness. We start by formalizing what it means mathematically for a

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

data value notion to be robust. We introduce the concept of *safety margin*, which is the magnitude of the largest perturbation of model performance scores that can be tolerated so that the value order of every pair of data points remains unchanged. We consider the two most popular data valuation schemes—the Shapley value and the Leave-one-out (LOO) error and show that the safety margin of the Shapley value is greater than that of the LOO error. Our results shed light on a common observation in the past works (Ghorbani and Zou, 2019; Jia et al., 2019c) that the Shapley value often outperforms the LOO error in identifying low-quality training data.

Banzhaf value: a robust data value notion. Surprisingly, we found that the Banzhaf value (Banzhaf III, 1964), a classic value notion from cooperative game theory that was proposed more than half a century ago, achieves the largest safety margin among all semivalues — a collection of value notions (including LOO error and the Shapley value) that satisfy essential properties of a proper data value notion in the ML context (Kwon and Zou, 2021). Particularly, the safety margin of the Banzhaf value is exponentially larger than that of the Shapley value and the LOO error.

Efficient Banzhaf value estimation algorithm. Similar to the Shapley value, the Banzhaf value is also costly in computation. We present an efficient estimation algorithm based on the Maximum Sample Reuse (MSR) principle, which can achieve ℓ_∞ and ℓ_2 error guarantees for approximating the Banzhaf value with logarithmic and nearly linear sample complexity, respectively. We show that the existence of an efficient MSR estimator is *unique* for the Banzhaf value among all existing semivalue-based data value notions. We derive a lower bound of sample complexity for the Banzhaf value estimation, and show that our MSR estimator's sample complexity *is close to* this lower bound. Additionally, we show that the MSR estimator is robust against the noise in performance scores.

Empirical evaluations. Our evaluation demonstrates the ability of the Banzhaf value in preserving value rankings given noisy model performance scores. We also empirically validate the sample efficiency of the MSR estimator for the Banzhaf value. We show that the Banzhaf value outperforms the state-of-the-art semivalue-based data value notions (including the Shapley value, the LOO error, and the recently proposed Beta Shapley (Kwon and Zou, 2021)) on several ML tasks including bad data detection and data reweighting, when the underlying learning algorithm is SGD.

We call the suite of our data value notion and the associated estimation algorithm as the *Data Banzhaf* framework. Overall, our work suggests that Data Banzhaf is a promising alternative to the existing semivalue-based data value notions given its computational advantage and the ability to robustly distinguish data quality in the presence of learning stochasticity.

2 BACKGROUND: FROM LOO TO SHAPLEY TO SEMIVALUE

In this section, we formalize the data valuation problem for ML. Then, we review the concept of LOO and Shapley value—the most popular data value notions in the existing literature, as well as the framework of *semivalues*, which are recently introduced as a natural relaxation of Shapley value in the ML context.

Data Valuation Problem Set-up. Let $N = \{1, \ldots, n\}$ denotes a training set of size n. The objective of data valuation is to assign a score to each training data point in a way that reflects their contribution to model training. We will refer to these scores as *data values*. To analyze a point's "contribution", we define a *utility function* $U : 2^N \to \mathbb{R}$, which maps any subset of the training set to a score indicating the usefulness of the subset. 2^N represents the power set of N, i.e., the collection of all subsets of N, including the empty set and N itself. For classification tasks, a common choice for U is the validation accuracy of a model trained on the input subset. Formally, we have $U(S) = \texttt{acc}(\mathcal{A}(S))$, where $\mathcal{A}$ is a learning algorithm that takes a dataset S as input and returns a model, and $\texttt{acc}$ is a metric function to evaluate the performance of a given model, e.g., the accuracy of a model on a hold-out test set. Without loss of generality, we assume throughout the paper that $U(S) \in [0, 1]$. For notational simplicity, we sometimes denote $S \cup i := S \cup \{i\}$ and $S \setminus i := S \setminus \{i\}$ for singleton $\{i\}$, where $i \in N$ represents a single data point.

We denote the data value of data point $i \in N$ computed from U as $\phi(i; U)$. We review the famous data value notions in the following.

LOO Error. A simple data value measure is leave-one-out (LOO) error, which calculates the change of model performance when the data point i is excluded from the training set N:

$$\phi_{\text{loo}}(i; U) := U(N) - U(N \setminus i) \tag{1}$$

However, many empirical studies (Ghorbani and Zou, 2019; Jia et al., 2019c) suggest that it underperforms other alternatives in differentiating data quality.

Shapley Value. The Shapley value is arguably the most widely studied scheme for data valuation. At a high level, it appraises each point based on the (weighted) average utility change caused by adding the point into different subsets. The Shapley value of a data point i is defined as

$$\phi_{\text{shap}}(i; U) := \frac{1}{n} \sum_{k=1}^{n} \binom{n-1}{k-1}^{-1} \sum_{S \subseteq N \setminus \{i\}, |S|=k-1} [U(S \cup i) - U(S)]$$

The popularity of the Shapley value is attributable to the

fact that it is the *unique* data value notion satisfying the following four axioms (Shapley, 1953):

- Dummy player: if $U(S \cup i) = U(S) + c$ for all $S \subseteq N \setminus i$ and some $c \in \mathbb{R}$, then $\phi(i; U) = c$.
- Symmetry: if $U(S \cup i) = U(S \cup j)$ for all $S \subseteq N \setminus \{i, j\}$, then $\phi(i; U) = \phi(j; U)$.
- Linearity: For utility functions U_1, U_2 and any $\alpha_1, \alpha_2 \in \mathbb{R}$, $\phi(i; \alpha_1 U_1 + \alpha_2 U_2) = \alpha_1 \phi(i; U_1) + \alpha_2 \phi(i; U_2)$.
- Efficiency: for every U, $\sum_{i \in N} \phi(i; U) = U(N)$.

The difference $U(S \cup i) - U(S)$ is often termed the *marginal contribution* of data point i to subset $S \subseteq N \setminus i$. We refer the readers to (Ghorbani and Zou, 2019; Jia et al., 2019b) for a detailed discussion about the interpretation of dummy player, symmetry, and linearity axioms in ML. The *efficiency* axiom, however, receives more controversy than the other three. The efficiency axiom requires the total sum of data values to be equal to the utility of full dataset $U(N)$. Recent work (Kwon and Zou, 2021) argues that this axiom is considered not essential in ML. Firstly, the choice of utility function in the ML context is often not directly related to monetary value so it is unnecessary to ensure the sum of data values matches the total utility. Moreover, many applications of data valuation, such as bad data detection, are performed based only on the ranking of data values. For instance, multiplying the Shapley value by a positive constant does not affect the ranking of the data values. Hence, there are many data values that do not satisfy the efficiency axiom, but can still be used for differentiating data quality, just like the Shapley value.

Semivalue. The class of data values that satisfy all the Shapley axioms except efficiency is called *semivalues*. It was originally studied in the field of economics and recently proposed to tackle the data valuation problem (Kwon and Zou, 2021). Unlike the Shapley value, semivalues are *not* unique. The following theorem by the seminal work of (Dubey et al., 1981) shows that every semivalue of a player i (in our case the player is a data point) can be expressed as the weighted average of marginal contributions $U(S \cup i) - U(S)$ across different subsets $S \subseteq N \setminus i$.

Theorem 2.1 (Representation of Semivalue (Dubey et al., 1981))**.** *A value function ϕ_{semi} is a semivalue, if and only if, there exists a* weight function $w : [n] \to \mathbb{R}$ *such that $\sum_{k=1}^{n} \binom{n-1}{k-1} w(k) = n$ and the value function ϕ_{semi} can be expressed as follows:*

$$\phi_{\text{semi}}(i; U, w) := \sum_{k=1}^{n} \frac{w(k)}{n} \sum_{\substack{S \subseteq N \setminus \{i\}, \\ |S| = k-1}} [U(S \cup i) - U(S)] \quad (2)$$

Semivalues subsume both the Shapley value and the LOO error with $w_{\text{shap}}(k) = \binom{n-1}{k-1}^{-1}$ and $w_{\text{loo}}(k) = n\mathbf{1}[k = n]$, respectively. Despite the theoretical attraction, the question remains which one of the many semivalues we should adopt.

3 UTILITY FUNCTIONS CAN BE STOCHASTIC

In the existing literature, the utility of a dataset $U(S)$ is often defined to be $\texttt{acc}(\mathcal{A}(S))$, i.e., the performance of a model $\mathcal{A}(S)$ trained on a dataset S. However, many learning algorithms $\mathcal{A}$ such as SGD contain randomness. Since the loss function for training neural networks is non-convex, the trained model depends on the randomness of the training process, e.g., random mini-batch selection. Thus, $U(S)$ defined in this way inherently becomes a randomized function. As noted in many studies on the reproducibility of neural network training, the learning stochasticity can introduce large variations into the predictive performance of deep learning models (Summers and Dinneen, 2021; Zhuang et al., 2022; Raste et al., 2022). On the other hand, the existing data value notions compute the value of data points based on the performance scores of models trained on different data subsets and therefore will also be noisy given stochastic learning algorithms. In this section, we delve into the influence of learning stochasticity on data valuation results, and show that the run-to-run variability of the resulting data value rankings is large for the existing data value notions.

Instability of data value rankings. Semivalues are calculated by taking a weighted average of marginal contributions. When the weights are not properly chosen, the noisy estimate of marginal contributions can cause significant instability in ranking the data values. Figure 1 (a)-(b) illustrate the distribution of the estimates of two popular data value notions—LOO error and the Shapley value—across 5 runs with different training random seeds. The utility function is the accuracy of a neural network trained via SGD on a held-out dataset; we show the box-plot of the estimates' distribution for 20 CIFAR10 images, with 5 of them being mislabeled (marked in red). The experiment settings are detailed in Appendix D.1. As we can see, the variance of the data value estimates caused by learning stochasticity significantly outweighs their magnitude for both LOO and the Shapley value. As a result, the rankings of data values across different runs are largely inconsistent (the average Spearman coefficient of individual points' values across different runs for LOO is ≈ 0.001 and for Shapley is ≈ 0.038). Leveraging the rankings of such data values to differentiate data quality is unreliable, as we can see that the 5 mislabeled images' value estimates distribution has a large overlap with the value estimates of the clean images. Further investigation of these data value notion's efficacy in identifying data quality is provided in the Evaluation section.

When interpreting a learning algorithm, one may be interested in finding a small set of data points with the most positive/negative influences on model performance. In Figure 2, we show how many data points are consistently ranked in the top or bottom-$k\%$ across all the runs. Both LOO and the Shapley value has only $< 10\%$ data points that are consis-

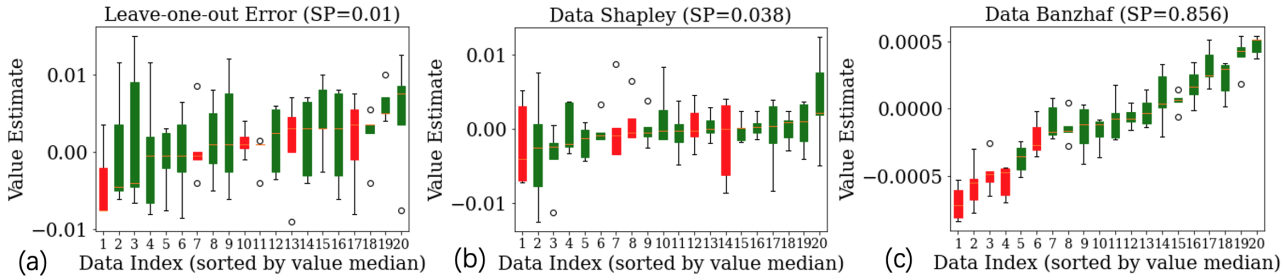

Figure 1: Box-plot of the estimates of (a) LOO, (b) Shapley Value, and (c) Banzhaf value of 20 randomly selected CIFAR10 images, with 5 mislabeled images. The 5 mislabeled images are shown in red and clean images are shown in green. The variance is *only* due to the stochasticity in utility evaluation. 'SP' means the average Spearman index across different runs.

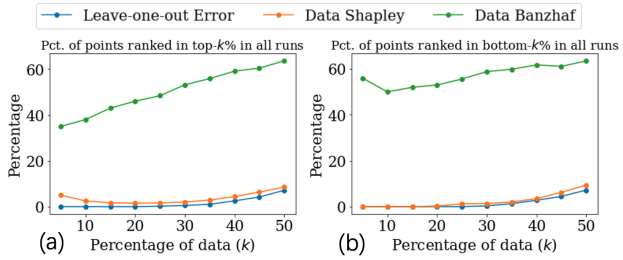

Figure 2: The percentage of the CIFAR10 data points that are ranked in (a) top-$k\%$ and (b) bottom-$k\%$ across all runs, among the top/bottom-$k\%$ data points.

tently ranked high/low for any $k \leq 50\%$. This means that the high/low-influence data points selected by these data value notions have a large run-to-run variation and cannot form a reliable explanation for model behaviors.

Redefine U as expected performance. To make the data value notions independent of the learning stochasticity, a natural choice is to redefine U to be $U(S) := \mathbb{E}_{\mathcal{A}}[\texttt{acc}(\mathcal{A}(S))]$, i.e., the *expected performance* of the trained model. However, accurately estimating $U(S)$ under this new definition requires running $\mathcal{A}$ multiple times on the same S, and calculating the average utility of S. Obviously, this simple approach incurs a large extra computational cost. On the other hand, if we estimate $U(S)$ with only one or few calls of $\mathcal{A}$, the estimate of $U(S)$ will be very noisy. Hence, we pose the question: *how to find a more robust semivalue against perturbation in model performance scores?*

4 DATA BANZHAF: A ROBUST DATA VALUE NOTION

To address the question posed above, this section starts by formalizing the notion of robustness in data valuation. Then, we show that the most robust semivalue, surprisingly, coincides with the Banzhaf value (Banzhaf III, 1964)—a classic solution concept in cooperative game theory.

4.1 Ranking Stability as a Robustness Notion

In many applications of data valuation such as data selection, it is the **order** of data values that matter (Kwon and Zou, 2021). For instance, to filter out low-quality data, one will first rank the data points based on their values and then throws the points with the lowest values. When the utility functions are perturbed by noise, we would like the rankings of the data values to remain stable. Recall that a semivalue is defined by a weight function w such that $\sum_{k=1}^{n} \binom{n-1}{k-1} w(k) = n$. The (scaled) difference between the semivalues of two data points i and j can be computed from (2):

$$D_{i,j}(U; w) := n(\phi(i; w) - \phi(j; w)) = \sum_{k=1}^{n-1} (w(k) + w(k+1)) \binom{n-2}{k-1} \Delta_{i,j}^{(k)}(U),$$

where $\Delta_{i,j}^{(k)}(U) := \binom{n-2}{k-1}^{-1} \sum_{|S|=k-1, S \subseteq N \setminus \{i,j\}} [U(S \cup i) - U(S \cup j)]$, representing the *average distinguishability between i and j on size-k sets using the noiseless utility function U*. Let $\widehat{U}$ denote a noisy estimate of U. We can see that $\widehat{U}$ and U produce different data value orders for i, j if and only if $D_{i,j}(U; w) D_{i,j}(\widehat{U}; w) \leq 0$.[1] An initial attempt to define robustness is in terms of the smallest amount of perturbation magnitude $\left\| \widehat{U} - U \right\|$ such that U and $\widehat{U}$ produce different data rankings.[2] However, such a definition is problematic due to its dependency on the original utility function U. If the noiseless U itself cannot sufficiently differentiate between i and j (i.e., $\Delta_{i,j}^{(k)}(U) \simeq 0$ for $k = 1, \ldots, n-1$), then $D_{i,j}(U; w)$ will be (nearly) zero and infinitesimal perturbation can switch the ranking of $\phi(i)$ and $\phi(j)$. To reasonably define the robustness of semivalues, we solely consider the collection of utility functions that can sufficiently "distinguish" between i and j.

[1] We note that when the two data points receive the same value, we usually break tie randomly, thus we use ≤ 0 instead of < 0.

[2] Here, we view the utility function U as a size-2^n vector where each entry corresponds to $U(S)$ of a subset $S \subseteq N$.

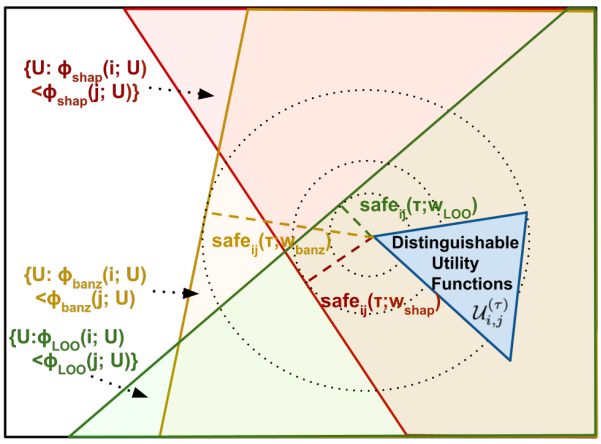

Figure 3: Illustration of Safety Margin. Intuitively, it is the smallest distance from the space of distinguishable utility function to the space of functions that will reverse the value rank between at least one pair of data points i and j. The figure shows the distance between the space of distinguishable utility function and the space of utility function where $\phi_{\text{banz}}(i; U) < \phi_{\text{banz}}(j; U)$ (i.e., the order of i, j being reversed) is the largest among the three value notions.

Definition 4.1 (Distinguishability)**.** We say a data point pair (i, j) is τ-*distinguishable* by U if and only if $\Delta_{i,j}^{(k)}(U) \geq \tau$ for all $k \in \{1, \ldots, n-1\}$.

Let $\mathcal{U}_{i,j}^{(\tau)}$ denote the collection of utility functions U that can τ-distinguish a pair (i, j). With the definition of distinguishability, we characterize the robustness of a semivalue by deriving its "*safety margin*", which is defined as the **minimum** amount of perturbation $\left\|\widehat{U} - U\right\|$ needed to reverse the ranking of **at least one** pair of data points (i, j), for **at least one** utility function U from $\mathcal{U}_{i,j}^{(\tau)}$.

Definition 4.2 (Safety margin)**.** Given $\tau > 0$, we define the safety margin of a semivalue for a data point pair (i, j) as

$$\texttt{Safe}_{i,j}(\tau; w) := \min_{U \in \mathcal{U}_{i,j}^{(\tau)}} \min_{\widehat{U} \in \{\widehat{U}: D_{i,j}(U;w) D_{i,j}(\widehat{U};w) \leq 0\}} \left\|\widehat{U} - U\right\|$$

and we define the safety margin of a semivalue as

$$\texttt{Safe}(\tau; w) := \min_{i,j \in N, i \neq j} \texttt{Safe}_{i,j}(\tau; w)$$

In other words, the safety margin captures the largest noise that can be tolerated by a semivalue without altering the ranking of any pair of data points that are distinguishable by the original utility function. The geometric intuition of safety margin is illustrated in Figure 3.

Remark 4.3. **The definition of the safety margin is *noise-structure-agnostic* in the sense that it does not depend on the actual noise distribution induced by a specific stochastic training algorithm.** While one might be tempted to have a noise-dependent robustness definition, we argue that the safety margin is advantageous from the following aspects: **(1) The analysis of the utility noise distribution caused by stochastic training is difficult even for very simple settings.** In Appendix B.2.1, we consider a simple (if not the simplest) setting: 1-dimensional linear regression trained by full-batch gradient descent with Gaussian random initialization. We show that, even for such a setting, there are significant technical challenges in making any analytical assertion on the noise distribution. **(2) It might be computationally infeasible to estimate the perturbation distribution of $U(S)$,** as there are exponentially many S that need to be considered. **(3) One may not have prior knowledge about the dataset and source of noise; in that case, a worst-case robustness notion is preferred.** In practice, the datasets may not be known in advance for privacy consideration (Agahari et al., 2022). Furthermore, the performance scores may also be perturbed due to other factors such as hardware faults, software bugs, or even adversarial attacks. Given the diversity and unknownness of utility perturbations in practical scenarios, it is preferable to define the robustness in terms of the worst-case perturbation (i.e., Kerckhoffs's principle). Such definitions are common in machine learning. For instance, robust learning against adversarial examples is aimed at being resilient to the worst-case noise (Madry et al., 2017) within a norm ball; differential privacy (Dwork et al., 2006) protects individual data record's information from arbitrary attackers.

A full discussion for important considerations in Definition 4.2 can be found in Appendix B.

Safety Margin for LOO and Shapley Value. In order to demonstrate the usefulness of this robustness notion, we derive the LOO and Shapley value's safety margin.

Theorem 4.4. *For any $\tau > 0$, Leave-one-out error ($w_{\text{loo}}(k) = n\mathbf{1}[k = n]$) achieves $\texttt{Safe}(\tau; w_{\text{loo}}) = \tau$, and Shapley value ($w_{\text{shap}}(k) = \binom{n-1}{k-1}^{-1}$) achieves $\texttt{Safe}(\tau; w_{\text{shap}}) = \tau \frac{n-1}{\sqrt{\sum_{k=1}^{n-1} \binom{n-2}{k-1}^{-1}}}$.*

One can easily see that $\texttt{Safe}(\tau; w_{\text{loo}}) < \texttt{Safe}(\tau; w_{\text{shap}}) < \tau(n-1)$. The fact that $\texttt{Safe}(\tau; w_{\text{loo}}) < \texttt{Safe}(\tau; w_{\text{shap}})$ sheds light on the phenomenon we observe in Figure 1 and 2 where the Shapley value is slightly more stable than LOO. It provides an explanation for a widely observed but puzzling phenomenon observed in several prior works (Jia et al., 2019c; Wang et al., 2020) that the Shapley value outperforms the LOO error in a range of data selection tasks in the stochastic learning setting. We note that the Shapley value is also shown to be better than LOO in deterministic learning (Ghorbani and Zou, 2019; Jia et al., 2019b), where theoretical underpinning is an open question.

4.2 Banzhaf Value Achieves the Largest Safety Margin

Surprisingly, the semivalue that achieves the largest safety margin coincides with the Banzhaf value, another famous value notion that averages the marginal contribution across all subsets. We first recall the definition of the Banzhaf value.

Definition 4.5 (Banzhaf III (1964))**.** The Banzhaf value for data point i is defined as

$$\phi_{\text{banz}}(i; U, N) := \frac{1}{2^{n-1}} \sum_{S \subseteq N \setminus i} [U(S \cup i) - U(S)] \quad (3)$$

The Banzhaf value is a semivalue, as we can recover its definition (3) from the general expression of semivalues (2) by setting the constant weight function $w(k) = \frac{n}{2^{n-1}}$ for all $k \in \{1, \ldots, n\}$. We then show our main result.

Theorem 4.6. *For any $\tau > 0$, Banzhaf value ($w(k) = \frac{n}{2^{n-1}}$) achieves the largest safety margin $\texttt{Safe}(\tau; w) = \tau 2^{n/2-1}$ among all semivalues.*

Intuition. The superior robustness of the Banzhaf value can be explained intuitively as follows: Semivalues assign different weights to the marginal contribution against different data subsets according to the weight function w. To construct a perturbation of the utility function that maximizes the influence on the corresponding semivalue, one needs to perturb the utility of the subsets that are assigned with higher weights. Hence, the best robustification strategy is to assign uniform weights to all subsets, which leads to the Banzhaf value. On the other hand, semivalues that assign heterogeneous weights to different subsets, such as the Shapley value and LOO error, suffer a lower safety margin.

Remark 4.7. **Banzhaf value is also the most robust semivalue in terms of the data value magnitude.** One can also show that the Banzhaf value is the most robust in the sense that the utility noise will minimally affect data value magnitude changes. Specifically, the Banzhaf value achieves the smallest Lipschitz constant L such that $\|\phi(U) - \phi(\widehat{U})\| \leq L\|U - \widehat{U}\|$ for all possible pairs of U and $\widehat{U}$. The details are deferred to Appendix C.4.

4.3 Efficient Banzhaf Value Estimation

Similar to the Shapley value and the other semivalue-based data value notions, the exact computation of the Banzhaf value can be expensive because it requires an exponential number of utility function evaluations, which entails an exponential number of model fittings. This could be a major challenge for adopting the Banzhaf value in practice. To address this issue, we present a novel Monte Carlo algorithm to approximate the Banzhaf value.

We start by defining the estimation error of an estimator. We say a semivalue estimator $\widehat{\phi}$ is an (ε, δ)-approximation to the true semivalue ϕ (in ℓ_p-norm) if and only if $\Pr_{\widehat{\phi}}[\|\phi - \widehat{\phi}\|_p \leq \varepsilon] \geq 1 - \delta$ where the randomness is over the execution of the estimator. For any data point pair (i, j), if $|\phi(i) - \phi(j)| \geq 2\varepsilon$, then an estimator that is (ε, δ)-approximation in ℓ_∞-norm is guaranteed to keep the data value order of i and j with probability at least $1 - \delta$.

Baseline: Simple Monte Carlo. The Banzhaf value can be equivalently expressed as follows:

$$\phi_{\text{banz}}(i) = \mathbb{E}_{S \sim \text{Unif}(2^{N \setminus i})} [U(S \cup i) - U(S)] \quad (4)$$

where $\text{Unif}(\cdot)$ to denote Uniform distribution over the power set of $N \setminus \{i\}$. Thus, a straightforward Monte Carlo (MC) method to estimate $\phi_{\text{banz}}(i)$ is to sample a collection of data subsets $\mathcal{S}_i$ from $2^{N \setminus i}$ uniformly at random, and then compute $\widehat{\phi}_{\text{MC}}(i) = \frac{1}{|\mathcal{S}_i|} \sum_{S \in \mathcal{S}_i} (U(S \cup i) - U(S))$. We can repeat the above procedure for each $i \in N$ and obtain the approximated semivalue vector $\widehat{\phi}_{\text{MC}} = [\widehat{\phi}_{\text{MC}}(1), \ldots, \widehat{\phi}_{\text{MC}}(n)]$. The sample complexity of this simple MC estimator can be bounded by Hoeffding's inequality.

Theorem 4.8. *$\widehat{\phi}_{\text{MC}}$ is an (ε, δ)-approximation to the exact Banzhaf value in ℓ_2-norm with $O(\frac{n^2}{\varepsilon^2} \log(\frac{n}{\delta}))$ calls of $U(\cdot)$, and in ℓ_∞-norm with $O(\frac{n}{\varepsilon^2} \log(\frac{n}{\delta}))$ calls of $U(\cdot)$.*

Proposed Algorithm: Maximum Sample Reuse (MSR) Monte Carlo. The simple MC method is sub-optimal since, for each sampled $S \in \mathcal{S}_i$, the value of $U(S)$ and $U(S \cup i)$ are only used for estimating $\phi_{\text{banz}}(i)$, i.e., the Banzhaf value of a single datum i. This inevitably results in a factor of n in the final sample complexity as we need the same amount of samples to estimate each $i \in N$. To address this weakness, we propose an advanced MC estimator which achieves *maximum sample reuse* (MSR). Specifically, by the linearity of expectation, we have $\phi_{\text{banz}}(i) = \mathbb{E}_{S \sim \text{Unif}(2^{N \setminus i})} [U(S \cup i)] - \mathbb{E}_{S \sim \text{Unif}(2^{N \setminus i})} [U(S)]$. Suppose we have m samples $\mathcal{S} = \{S_1, \ldots, S_m\}$ i.i.d. drawn from $\text{Unif}(2^N)$. For every data point i we can divide $\mathcal{S}$ into $\mathcal{S}_{\ni i} \cup \mathcal{S}_{\not\ni i}$ where $\mathcal{S}_{\ni i} = \{S \in \mathcal{S} : i \in S\}$ and $\mathcal{S}_{\not\ni i} = \{S \in \mathcal{S} : i \notin S\} = \mathcal{S} \setminus \mathcal{S}_{\ni i}$. We can then estimate $\phi(i)$ by

$$\widehat{\phi}_{\text{MSR}}(i) = \frac{1}{|\mathcal{S}_{\ni i}|} \sum_{S \in \mathcal{S}_{\ni i}} U(S) - \frac{1}{|\mathcal{S}_{\not\ni i}|} \sum_{S \in \mathcal{S}_{\not\ni i}} U(S) \quad (5)$$

or set $\widehat{\phi}_{\text{MSR}}(i) = 0$ if either of $|\mathcal{S}_{\ni i}|$ and $|\mathcal{S}_{\not\ni i}|$ is 0. In this way, the maximal sample reuse is achieved since *all* evaluations of $U(S)$ are used in the estimation of $\phi(i)$ for every $i \in N$. We refer to this new estimator $\widehat{\phi}_{\text{MSR}}$ as *MSR estimator*. Compared with Simple MC method, the MSR estimator saves a factor of n in the sample complexity.

Theorem 4.9. *$\widehat{\phi}_{\text{MSR}}$ is an (ε, δ)-approximation to the exact Banzhaf value in ℓ_2-norm with $O\left(\frac{n}{\varepsilon^2} \log(\frac{n}{\delta})\right)$ calls of $U(\cdot)$, and in ℓ_∞-norm with $O\left(\frac{1}{\varepsilon^2} \log(\frac{n}{\delta})\right)$ calls of $U(\cdot)$.*

Proof overview. We remark that deriving this sample complexity is non-trivial. Unlike the simple Monte Carlo, the

sizes of the samples that we average over in (5) (i.e., $|\mathcal{S}_{\ni i}|$ and $|\mathcal{S}_{\not\ni i}|$) are also random variables. Hence, we cannot simply apply Hoeffding's inequality to get a high-probability bound for $\|\widehat{\phi}_{\mathrm{MSR}} - \phi_{\mathrm{banz}}\|$. The key to the proof is to notice that $|\mathcal{S}_{\ni i}|$ follows binomial distribution $\mathrm{Bin}(m, 0.5)$. Thus, we first show that $|\mathcal{S}_{\ni i}|$ is close to $m/2$ with high probability, and then apply Hoeffding's inequality to bound the difference between $\frac{1}{m/2}\sum_{S\in\mathcal{S}_{\ni i}} U(S) - \frac{1}{m/2}\sum_{S\in\mathcal{S}_{\not\ni i}} U(S)$ and $\phi_{\mathrm{banz}}(i)$.

The actual estimator of the Banzhaf value that we build is based upon the noisy variant $\widehat{U}$. In Appendix C.3, we study the impact of noisy utility function evaluation on the sample complexity of the MSR estimator. It can be shown that our MSR algorithm has the same sample complexity with the noisy $\widehat{U}$, despite a small extra irreducible error.

The existence of an efficient MSR estimator is unique for the Banzhaf value. Every semivalue can be written as the expectation of *weighted* marginal contribution. Hence, one could construct an MSR estimator for arbitrary semivalue as follows: $\widehat{\phi}_{\mathrm{MSR}}(i) = \frac{2^{n-1}}{n|\mathcal{S}_{\ni i}|}\sum_{S\in\mathcal{S}_{\ni i}} w(|S|)U(S) - \frac{2^{n-1}}{n|\mathcal{S}_{\not\ni i}|}\sum_{S\in\mathcal{S}_{\not\ni i}} w(|S|+1)U(S)$. For the Shapley value, $w(|S|) = \binom{n-1}{|S|-1}^{-1}$. This combinatorial coefficient makes the calculation of this estimator numerically unstable when n is large. As we will show in the Appendix C.2, it turns out that it is also impossible to construct a distribution $\mathcal{D}$ over 2^N s.t. $\phi_{\mathrm{shap}}(i) = \mathbb{E}_{S\sim\mathcal{D}|\mathcal{D}\ni i}[U(S)] - \mathbb{E}_{S\sim\mathcal{D}|\mathcal{D}\not\ni i}[U(S)]$ for the Shapley value and *any* other data value notions except the Banzhaf value. Hence, *the existence of the efficient MSR estimator is a unique advantage of the Banzhaf value*.

Lower Bound for Banzhaf Value Estimation. To understand the optimality of the MSR estimator, we derive a lower bound for any Banzhaf estimator that achieves (ε, δ)-approximation in ℓ_∞-norm. The main idea of deriving the lower bound is to use Yao's minimax principle. Specifically, we construct a distribution over instances of utility functions and prove that no deterministic algorithm can work well against that distribution.

Theorem 4.10. *Every (randomized) Banzhaf value estimator that achieves (ε, δ)-approximation in ℓ_∞-norm for constant $\delta \in (0, 1/2)$ has sample complexity at least $\Omega(\frac{1}{\varepsilon})$.*

Recall that our MSR algorithm achieves $\widetilde{O}(\frac{1}{\varepsilon^2})$[3] sample complexity. This means that our MSR algorithm is close to optimal, with an extra factor of $\frac{1}{\varepsilon}$.

5 EVALUATION

Our evaluation covers the following aspects: **(1)** Sample efficiency of the proposed MSR estimator for the Banzhaf value; **(2)** Robustness of the Banzhaf value compared to the six existing semivalue-based data value notions (including Shapley value, LOO error, and four representatives from Beta Shapley[4]); **(3)** Effectiveness of performing *noisy label detection* and *learning with weighted samples* based on the Banzhaf value. Detailed settings are provided in Appendix D.

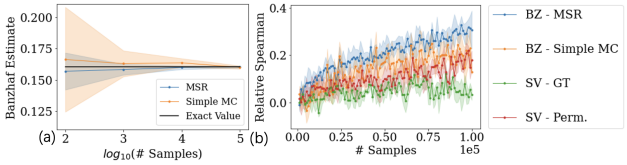

Figure 4: (a) Convergence of the MSR and simple MC estimator for the Banzhaf value of a data point from a synthetic dataset. The shaded area means the estimation variance over 5 runs of sampling. (b) The Relative Spearman Index of the estimators for Shapley and Banzhaf value on the MNIST dataset. 'SV-GT' refers to Group Testing-based Shapley estimator, and 'SV-Perm.' refers to Permutation Sampling Shapley estimator. For 'SV-GT', we implement the improved Group Testing-based Shapley estimator proposed in Wang and Jia (2023) instead of the original version from Jia et al. (2019b).

5.1 Sample Efficiency

MSR vs. Simple MC. We compare the sample complexity of the MSR and the simple MC estimator for approximating the Banzhaf value. In order to exactly evaluate the estimation error of the two estimators, we use a synthetic dataset generated by multivariate Gaussian with only 10 data points—a scale where we can compute the Banzhaf value exactly. The utility function is the validation accuracy of logistic regression trained with full-batch gradient descent (no randomness in training). Thus, the randomness associated with the estimator error is solely from random sampling in the estimation algorithm. Figure 4 (a) compares the variance of the two estimators as the number of samples grows. As we can see, the estimation error of the MSR estimator reduces much more quickly than that of the simple MC estimator. Furthermore, given the same amount of samples, the MSR estimator exhibits a much smaller variance across different runs compared to the simple MC method.

Banzhaf vs. Shapley. We then compare the two Banzhaf value estimators with two popular Shapley value estimators: the *Permutation Sampling* (Castro et al., 2009) and the *Group Testing* algorithm (Jia et al., 2019b; Wang and Jia, 2023). Since the Shapley and Banzhaf values are of different scales, for a fair comparison, we measure the consistency of the ranking of estimated data values. Specifically, we increase the sample size by adding a new batch of samples

[3] Throughout the paper, we use $\widetilde{O}$ to hide logarithmic factors.

[4] We evaluate Beta(1, 4), Beta(1, 16), Beta(4, 1), Beta(16, 1) as the original paper.

Dataset	Data Banzhaf	LOO	Beta(16, 1)	Beta(4, 1)	Data Shapley	Beta(1, 4)	Beta(1, 16)	Uniform
MNIST	**0.745 (0.026)**	0.708 (0.04)	-	-	0.74 (0.029)	-	-	0.733 (0.021)
FMNIST	**0.591 (0.014)**	0.584 (0.02)	-	-	0.581 (0.017)	-	-	0.586 (0.013)
CIFAR10	**0.642 (0.002)**	0.618 (0.005)	-	-	0.635 (0.004)	-	-	0.609 (0.004)
Click	**0.6 (0.002)**	0.575 (0.005)	-	-	0.589 (0.002)	-	-	0.57 (0.005)
Fraud	**0.923 (0.002)**	0.907 (0.002)	0.912 (0.004)	0.919 (0.005)	0.899 (0.002)	0.897 (0.001)	0.897 (0.001)	0.906 (0.002)
Creditcard	**0.66 (0.002)**	0.637 (0.006)	0.646 (0.003)	0.658 (0.007)	0.654 (0.003)	0.643 (0.004)	0.629 (0.007)	0.632 (0.003)
Vehicle	**0.814 (0.003)**	0.792 (0.008)	0.796 (0.003)	0.806 (0.004)	0.808 (0.003)	0.805 (0.005)	0.8 (0.004)	0.791 (0.005)
Apsfail	0.925 (0.0)	0.921 (0.003)	0.924 (0.001)	**0.926 (0.001)**	0.921 (0.002)	0.92 (0.002)	0.919 (0.001)	0.921 (0.002)
Phoneme	**0.778 (0.001)**	0.766 (0.006)	0.765 (0.002)	0.766 (0.005)	0.77 (0.004)	0.767 (0.003)	0.766 (0.003)	0.758 (0.002)
Wind	**0.832 (0.003)**	0.828 (0.002)	0.827 (0.003)	0.831 (0.002)	0.825 (0.002)	0.823 (0.002)	0.823 (0.002)	0.825 (0.003)
Pol	**0.856 (0.005)**	0.834 (0.008)	0.837 (0.009)	0.848 (0.004)	0.836 (0.014)	0.824 (0.007)	0.812 (0.008)	0.841 (0.009)
CPU	0.896 (0.001)	0.897 (0.002)	**0.899 (0.001)**	0.897 (0.002)	0.894 (0.002)	0.892 (0.001)	0.889 (0.002)	0.895 (0.001)
2DPlanes	**0.846 (0.006)**	0.83 (0.006)	0.837 (0.006)	0.841 (0.003)	**0.846 (0.005)**	0.843 (0.006)	0.838 (0.007)	0.829 (0.007)

Table 1: Accuracy comparison of models trained with weighted samples. We compare the seven data valuation methods on the 13 classification datasets. For MNIST, FMNIST, and CIFAR10 we use a size-2000 subset. The average and standard error of classification accuracy are denoted by 'average (standard error)'. The standard error is only due to the stochasticity in utility function evaluation. Boldface numbers denote the best method. Beta Shapley does NOT applicable for datasets with ≥ 1000 data points (MNIST, FMNIST, CIFAR10, and Click) due to numerical instability. 'Uniform' denotes training with uniformly weighted samples.

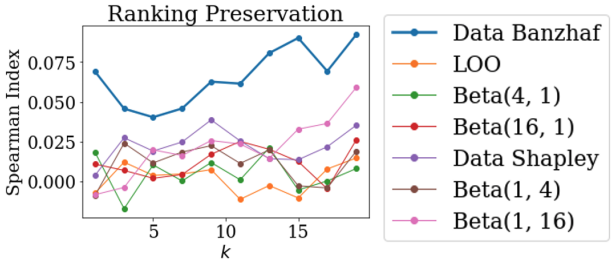

Figure 5: Impact of the noise in utility scores on the Spearman index between the ranking of reference data value and the ranking of data value estimated from noisy utility scores. The 'k' in x-axis means the number of repeated evaluations of $U(S)$. **The larger the k, the smaller the noise magnitude.** Detailed experiment procedure are in Appendix D.4.

at every iteration, and evaluate each of the estimators on different sample sizes. For each estimator, we calculate the **Relative Spearman Index**, which is the Spearman index of the value estimates between two adjacent iterations. A high Relative Spearman Index means the ranking does not change too much with extra samples, which implies convergence of data value rankings. Figure 4 (b) compares the Relative Spearman Index of different data value estimators on MNIST dataset. We can see that the MSR estimator for the Banzhaf value converges much faster than the two estimators for the Shapley value in terms of data value rankings.

5.2 Ranking Stability under Noisy Utility Functions

We compare the robustness of data value notions in preserving the value ranking against the utility score perturbation due to the stochasticity in SGD. The tricky part in the experiment design is that *we need to adjust the scale of the perturbation caused by natural stochastic learning algorithm*. In Appendix D.4, we show a procedure for controlling the magnitude of perturbation via repeatedly evaluating $U(S)$ for k times: the larger the k, the smaller the noise magnitude in the averaged $\widetilde{U}(S)$. In Figure 5, we plot the Spearman index between the ranking of reference data values[5] and the ranking of data values estimated from noisy utility scores. Detailed settings are in Appendix D.4. As we can see, Data Banzhaf achieves the most stable ranking and its stability advantage gets more prominent as the noise increases. Moreover, we show the box-plot of Banzhaf value estimates in Figure 1 (c). Compared with LOO and Shapley value, learning stochasticity has a much smaller impact on the ranking of Banzhaf values (the average Spearman index ≈ 0.856 compared with the one for Shapley value ≈ 0.038). Back to Figure 2, the high/low-quality data points selected by the Banzhaf value is much more consistent across different runs compared with LOO and Shapley value.

5.3 Applications of Data Banzhaf

Given the promising results obtained from the proof-of-concept evaluation in Section 5.1 and 5.2, we move forward to real-world applications and evaluate the effectiveness of Data Banzhaf in distinguishing data quality for machine learning tasks. Particularly, we considered two applications enabled by data valuation: one is to reweight training data during learning and another is to detect mislabeled points. We use neural networks trained with Adam as the learning algorithm wherein the associated utility function is noisy in nature. We remark that most prior works (e.g., Ghorbani and Zou (2019) and Kwon and Zou (2021)) use deterministic Logistic Regression to avoid the randomness in data

[5] We approximate the ground-truth by setting $k = 50$.

Dataset	Data Banzhaf	LOO	Beta(16, 1)	Beta(4, 1)	Data Shapley	Beta(1, 4)	Beta(1, 16)
MNIST	**0.193 (0.017)**	0.165 (0.009)	-	-	0.135 (0.025)	-	-
FMNIST	0.156 (0.018)	**0.164 (0.014)**	-	-	0.135 (0.016)	-	-
CIFAR10	**0.22 (0.003)**	0.086 (0.02)	-	-	0.152 (0.023)	-	-
Click	**0.206 (0.01)**	0.096 (0.034)	-	-	0.116 (0.024)	-	-
Fraud	0.47 (0.024)	0.157 (0.046)	0.55 (0.032)	0.59 (0.037)	**0.65 (0.032)**	0.19 (0.058)	0.14 (0.058)
Creditcard	0.27 (0.024)	0.113 (0.073)	0.25 (0.063)	**0.28 (0.081)**	0.26 (0.049)	0.17 (0.024)	0.17 (0.087)
Vehicle	**0.45 (0.0)**	0.123 (0.068)	0.43 (0.051)	0.42 (0.068)	0.41 (0.066)	0.16 (0.058)	0.1 (0.055)
Apsfail	**0.49 (0.037)**	0.096 (0.09)	0.36 (0.02)	0.42 (0.024)	0.47 (0.024)	0.22 (0.051)	0.2 (0.071)
Phoneme	0.216 (0.023)	0.115 (0.026)	0.232 (0.02)	**0.236 (0.027)**	0.216 (0.032)	0.124 (0.039)	0.088 (0.02)
Wind	0.36 (0.02)	0.073 (0.022)	0.51 (0.037)	0.52 (0.04)	**0.57 (0.068)**	0.19 (0.086)	0.17 (0.06)
Pol	**0.47 (0.04)**	0.097 (0.093)	0.26 (0.037)	0.4 (0.055)	0.44 (0.058)	0.17 (0.051)	0.09 (0.02)
CPU	0.35 (0.045)	0.107 (0.074)	0.45 (0.055)	**0.48 (0.06)**	0.46 (0.037)	0.13 (0.068)	0.08 (0.081)
2DPlanes	0.422 (0.025)	0.153 (0.057)	0.338 (0.034)	0.471 (0.041)	**0.512 (0.082)**	0.471 (0.041)	0.338 (0.034)

Table 2: Comparison of mislabel data detection ability of the seven data valuation methods on the 13 classification datasets. The average and standard error of F1-score are denoted by 'average (standard error)'. The standard error is only due to the random noise in the utility function evaluation. Boldface numbers denote the best method in F1-score average.

value results. We compare with 6 baselines that are previously proposed semivalue-based data value notions: Data Shapley, Leave-one-out (LOO), and 4 variations of Beta Shapley (Kwon and Zou, 2021) (Beta(1, 4), Beta(1, 16), Beta(4, 1), Beta(16, 1)).[6] We use 13 standard datasets that are previously used in the data valuation literature as the benchmark tasks.

Learning with Weighted Samples. Similar to Kwon and Zou (2021), we weight each training point by normalizing the associated data value between [0,1]. Then, during training, each training sample will be selected with a probability equal to the assigned weight. As a result, data points with a higher value are more likely to be selected in the random mini-batch of SGD, and data points with a lower value are rarely used. We train a neural network classifier to minimize the weighted loss, and then evaluate the accuracy on the held-out test dataset. As Table 1 shows, Data Banzhaf outperforms other baselines.

Noisy Label Detection. We investigate the ability of different data value notions in detecting mislabeled points under noisy utility functions. We generate noisy labeled samples by flipping labels for a randomly chosen 10% of training data points. We mark a data point as a mislabeled one if its data value is less than 10 percentile of all data value scores. We use F1-score as the performance metric for mislabeling detection. Table 2 in the Appendix shows the F1-score of the 7 data valuation methods and Data Banzhaf shows the best overall performance.

[6]Beta Shapley does not apply for datasets with $\geq$ 1000 data points due to numerical instability.

6 LIMITATION AND FUTURE WORK

This work presents the first focused study on the reliability of data valuation in the presence of learning stochasticity. We develop Data Banzhaf, a new data valuation method that is more robust against the perturbations to the model performance scores than existing ones. One limitation of this study is that the robustness guarantee is a worst-case guarantee, which assumes the perturbation could be arbitrary or even adversarial. While such a robustness notion is advantageous in many aspects as discussed in Remark 4.3, the threat model might be strong when the source of perturbation is known. Developing new robustness notions which take into account specific utility noise distributions induced by learning stochasticity is important future work, and it requires a breakthrough in understanding the noise structure caused by learning stochasticity.

Acknowledgments

The work was supported by Princeton's Gordon Y. S. Wu Fellowship and Cisco Research Awards. We thank Sanjeev Arora for the helpful discussion. We thank Yongchan Kwon for sharing the implementation of Beta Shapley and dataset sources. We thank Rachel Li and Jacqueline Wei for sharing the summary of the discussion about this work in CS236r (Topics at the Interface between Computer Science and Economics, Fall 2022) at Harvard. We are grateful to anonymous reviewers at AISTATS for their valuable feedback.

References

Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55. US Government printing office, 1964.

Wirawan Agahari, Hosea Ofe, and Mark de Reuver. It is not (only) about privacy: How multi-party computation redefines control, trust, and risk in data sharing. *Electronic Markets*, pages 1–26, 2022.

Lucas Agussurja, Xinyi Xu, and Bryan Kian Hsiang Low. On the convergence of the shapley value in parametric bayesian learning games. *arXiv preprint arXiv:2205.07428*, 2022.

Mohammad Mohammadi Amiri, Frederic Berdoz, and Ramesh Raskar. Fundamentals of task-agnostic data valuation. *arXiv preprint arXiv:2208.12354*, 2022.

Haris Aziz. Complexity of comparison of influence of players in simple games. *arXiv preprint arXiv:0809.0519*, 2008.

Yoram Bachrach, Evangelos Markakis, Ezra Resnick, Ariel D Procaccia, Jeffrey S Rosenschein, and Amin Saberi. Approximating power indices: theoretical and empirical analysis. *Autonomous Agents and Multi-Agent Systems*, 20(2):105–122, 2010.

John F Banzhaf III. Weighted voting doesn't work: A mathematical analysis. *Rutgers L. Rev.*, 19:317, 1964.

Yatao Bian, Yu Rong, Tingyang Xu, Jiaxiang Wu, Andreas Krause, and Junzhou Huang. Energy-based learning for cooperative games, with applications to valuation problems in machine learning. *arXiv preprint arXiv:2106.02938*, 2021.

Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the shapley value based on sampling. *Computers & Operations Research*, 36(5):1726–1730, 2009.

James Coleman. Control of collectivities and the power of a collectivity to act, in, b. lieberman (ed.), social choice, 1971.

Ian C Covert, Scott Lundberg, and Su-In Lee. Shapley feature utility.

Andrea Dal Pozzolo, Olivier Caelen, Reid A Johnson, and Gianluca Bontempi. Calibrating probability with undersampling for unbalanced classification. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 159–166. IEEE, 2015.

Abhranil Das and Wilson S Geisler. A method to integrate and classify normal distributions. *Journal of Vision*, 21 (10):1–1, 2021.

Amit Datta, Anupam Datta, Ariel D Procaccia, and Yair Zick. Influence in classification via cooperative game theory. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

Robert B Davies. Algorithm as 155: The distribution of a linear combination of χ 2 random variables. *Applied Statistics*, pages 323–333, 1980.

John Deegan and Edward W Packel. A new index of power for simplen-person games. *International Journal of Game Theory*, 7(2):113–123, 1978.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Xiaotie Deng and Christos H Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of operations research*, 19(2):257–266, 1994.

Marco F Duarte and Yu Hen Hu. Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):826–838, 2004.

Pradeep Dubey, Abraham Neyman, and Robert James Weber. Value theory without efficiency. *Mathematics of Operations Research*, 6(1):122–128, 1981.

John C Duchi, Peter L Bartlett, and Martin J Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

Shaheen S Fatima, Michael Wooldridge, and Nicholas R Jennings. A linear approximation method for the shapley value. *Artificial Intelligence*, 172(14):1673–1699, 2008.

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

Amirata Ghorbani, Michael Kim, and James Zou. A distributional framework for data valuation. In *International Conference on Machine Learning*, pages 3535–3544. PMLR, 2020.

Amirata Ghorbani, James Zou, and Andre Esteva. Data shapley valuation for efficient batch active learning. *arXiv preprint arXiv:2104.08312*, 2021.

Dongge Han, Michael Wooldridge, Alex Rogers, Shruti Tople, Olga Ohrimenko, and Sebastian Tschiatschek. Replication-robust payoff-allocation for machine learning data markets. *arXiv preprint arXiv:2006.14583*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Manfred J Holler. Forming coalitions and measuring voting power. *Political studies*, 30(2):262–271, 1982.

Manfred J Holler and Edward W Packel. Power, luck and the right index. *Zeitschrift für Nationalökonomie*, 43(1): 21–29, 1983.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas J Spanos, and Dawn Song. Efficient task-specific data valuation for nearest neighbor algorithms. *arXiv preprint arXiv:1908.08619*, 2019a.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas 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, 2019b.

Ruoxi Jia, Fan Wu, Xuehui Sun, Jiacen Xu, David Dao, Bhavya Kailkhura, Ce Zhang, Bo Li, and Dawn Song. Scalability vs. utility: Do we have to sacrifice one for the other in data importance quantification? *arXiv preprint arXiv:1911.07128*, 2019c.

Adam Karczmarz, Anish Mukherjee, Piotr Sankowski, and Piotr Wygocki. Improved feature importance computations for tree models: Shapley vs. banzhaf. *arXiv preprint arXiv:2108.04126*, 2021.

Bojan Karlaš, David Dao, Matteo Interlandi, Bo Li, Sebastian Schelter, Wentao Wu, and Ce Zhang. Data debugging with shapley importance over end-to-end machine learning pipelines. *arXiv preprint arXiv:2204.11131*, 2022.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Bogdan Kulynych and Carmela Troncoso. Feature importance scores and lossless feature pruning using banzhaf power indices. *arXiv preprint arXiv:1711.04992*, 2017.

Yongchan Kwon and James Zou. Beta shapley: a unified and noise-reduced data valuation framework for machine learning. *arXiv preprint arXiv:2110.14049*, 2021.

Yongchan Kwon, Manuel A Rivas, and James Zou. Efficient computation and analysis of distributional shapley values. In *International Conference on Artificial Intelligence and Statistics*, pages 793–801. PMLR, 2021.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989.

Jinkun Lin, Anqi Zhang, Mathias Lécuyer, Jinyang Li, Aurojit Panda, and Siddhartha Sen. Measuring the effect of training data on deep learning predictions via randomized experiments. In *International Conference on Machine Learning*, pages 13468–13504. PMLR, 2022.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Sasan Maleki. *Addressing the computational issues of the Shapley value with applications in the smart grid*. PhD thesis, University of Southampton, 2015.

Irwin Mann and Lloyd S Shapley. *Values of large games, IV: Evaluating the electoral college by Montecarlo techniques*. Rand Corporation, 1960.

Samuel Merrill III. Approximations to the banzhaf index of voting power. *The American Mathematical Monthly*, 89 (2):108–110, 1982.

Guillermo Owen. Multilinear extensions of games. *Management Science*, 18(5-part-2):64–79, 1972.

Neel Patel, Martin Strobel, and Yair Zick. High dimensional model explanations: An axiomatic approach. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 401–411, 2021.

Lionel S Penrose. The elementary statistics of majority voting. *Journal of the Royal Statistical Society*, 109(1): 53–57, 1946.

Soham Raste, Rahul Singh, Joel Vaughan, and Vijayan N Nair. Quantifying inherent randomness in machine learning algorithms. *arXiv preprint arXiv:2206.12353*, 2022.

Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

Rachael Hwee Ling Sim, Yehong Zhang, Mun Choon Chan, and Bryan Kian Hsiang Low. Collaborative machine learning with incentive-aware model rewards. In *International Conference on Machine Learning*, pages 8927–8936. PMLR, 2020.

Rachael Hwee Ling Sim, Xinyi Xu, and Bryan Kian Hsiang Low. Data valuation in machine learning:"ingredients", strategies, and open challenges. In *Proc. IJCAI*, 2022.

Jakub Sliwinski, Martin Strobel, and Yair Zick. Axiomatic characterization of data-driven influence measures for classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 718–725, 2019.

Cecilia Summers and Michael J Dinneen. Nondeterminism and instability in neural network optimization. In *International Conference on Machine Learning*, pages 9913–9922. PMLR, 2021.

Siyi Tang, Amirata Ghorbani, Rikiya Yamashita, Sameer Rehman, Jared A Dunnmon, James Zou, and Daniel L Rubin. Data valuation for medical imaging using shapley value and application to a large-scale chest x-ray dataset. *Scientific reports*, 11(1):1–9, 2021.

Sebastian Shenghong Tay, Xinyi Xu, Chuan Sheng Foo, and Bryan Kian Hsiang Low. Incentivizing collaboration in machine learning via synthetic data rewards. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9448–9456, 2022.

Jacopo Teneggi, Alexandre Luster, and Jeremias Sulam. Fast hierarchical games for image explanations. *arXiv preprint arXiv:2104.06164*, 2021.

Zhihua Tian, Jian Liu, Jingyu Li, Xinle Cao, Ruoxi Jia, and Kui Ren. Private data valuation and fair payment in data marketplaces. *arXiv preprint arXiv:2210.08723*, 2022.

Jiachen T. Wang and Ruoxi Jia. A note on "towards efficient data valuation based on the shapley value", 2023.

Tianhao Wang, Johannes Rausch, Ce Zhang, Ruoxi Jia, and Dawn Song. A principled approach to data valuation for federated learning. In *Federated Learning*, pages 153–167. Springer, 2020.

Tianhao Wang, Yu Yang, and Ruoxi Jia. Improving cooperative game theory-based data valuation via data utility learning. *arXiv preprint arXiv:2107.06336*, 2021.

Robert J Weber. Probabilistic values for games. *The Shapley Value. Essays in Honor of Lloyd S. Shapley*, pages 101–119, 1988.

Zhaoxuan Wu, Yao Shu, and Bryan Kian Hsiang Low. Davinz: Data valuation using deep neural networks at initialization. In *International Conference on Machine Learning*, pages 24150–24176. PMLR, 2022.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Xinyi Xu, Zhaoxuan Wu, Chuan Sheng Foo, and Bryan Kian Hsiang Low. Validation free and replication robust volume-based data valuation. *Advances in Neural Information Processing Systems*, 34:10837–10848, 2021.

Tom Yan and Ariel D Procaccia. If you like shapley then you'll love the core, 2020.

I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert systems with applications*, 36(2):2473–2480, 2009.

Gal Yona, Amirata Ghorbani, and James Zou. Who's responsible? jointly quantifying the contribution of the learning algorithm and data. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 1034–1041, 2021.

Jinsung Yoon, Sercan Arik, and Tomas Pfister. Data valuation using reinforcement learning. In *International Conference on Machine Learning*, pages 10842–10851. PMLR, 2020.

Liehuang Zhu, Hui Dong, Meng Shen, and Keke Gai. An incentive mechanism using shapley value for blockchain-based medical data sharing. In *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, pages 113–118. IEEE, 2019.

Donglin Zhuang, Xingyao Zhang, Shuaiwen Song, and Sara Hooker. Randomness in neural network training: Characterizing the impact of tooling. *Proceedings of Machine Learning and Systems*, 4:316–336, 2022.

A EXTENDED RELATED WORK

Cooperative Game Theory-based Data Valuation. Game-theoretic formulations of data valuation have become popular in recent years due to the formal, axiomatic justification. Particularly, the Shapley value has become a popular data value notion (Ghorbani and Zou, 2019; Jia et al., 2019b) as it is the unique value notion that satisfies the four axioms: *linearity, dummy player, symmetry, and efficiency*. Alternatives to the Shapley value for data valuation have also been proposed through the relaxation of the Shapley axioms. As we mentioned in the main text, by relaxing the efficiency axiom, the class of solution concepts that satisfy linearity, dummy player, and symmetry is called *semivalue* (Weber, 1988). Kwon and Zou (2021) propose *Beta Shapley*, which is a collection of semivalues that enjoy certain mathematical elegance in the representation. However, the construction of Beta Shapley does *not* take the perturbation of performance scores into account; the original paper only uses deterministic learning algorithms (such as Logistic Regression) for the experiment. In this work, we characterize the Banzhaf value as a robust semivalue against the perturbation in the performance scores, which is critical for applications involving stochastic training such as neural network training.

In addition, by relaxing the linearity axiom of the Shapley value, Yan and Procaccia (2020) propose to use the *Least core* (Deng and Papadimitriou, 1994), another classic concept in cooperative game theory, as an alternative to the Shapley value for data valuation. At a high level, the Least Core is a profit allocation scheme that requires the smallest subsidy to each coalition S so that no participant has the incentive to deviate from the grand coalition N. It is computed by solving the linear programming problem below:

$$\min_{\phi_{\mathrm{LC}}} e \quad \text{s.t. } \sum_{i=1}^{n} \phi_{\mathrm{LC}}(i) = U(N), \quad \sum_{i \in S} \phi_{\mathrm{LC}}(i) + e \geq U(S), \forall S \subseteq N \tag{6}$$

We show additional experiment results for the robustness of the Least core in Appendix D.4.

Distributional Shapley value (Ghorbani et al., 2020; Kwon et al., 2021) is a variant of Data Shapley which measures the contribution of a data point with respect to a *data distribution* instead of a static dataset. The stability notion discussed in the paper is in terms of the perturbation to the data point instead of model performance scores. Bian et al. (2021) take a probabilistic treatment of cooperative games. Through mean-field variational inference in the energy-based model, they develop *multiple step variational value* as a data value notion that satisfies null player, marginalism, and symmetry. The marginalism axiom requires a player's payoffs to depend only on his own marginal contributions – whenever they remain unchanged, his payoffs should be unaffected. Yona et al. (2021) relax the assumption that the learning algorithm is fixed in advance in the previous work, and extend Shapley value to jointly quantify the contribution of data points and learning algorithms. It improves the stability of data value under domain shifts by attributing the responsibility to the learning algorithm. Agussurja et al. (2022) derive the convergence property of the Shapley value in parametric Bayesian learning games, and apply the result to establish an online collaborative learning framework that is asymptotically Shapley-fair.

Banzhaf value, Banzhaf power index, and friends. What is known today as the Banzhaf value or Banzhaf power index was originally introduced by Lionel Penrose in 1946 (Penrose, 1946). It was reinvented by John F. Banzhaf III in 1964 (Banzhaf III, 1964), and was reinvented once more by James Samuel Coleman in 1971 (Coleman, 1971) before it became part of the mainstream literature. In the field of machine learning, the Banzhaf value has been previously applied to the problem of measuring feature importance (Datta et al., 2015; Kulynych and Troncoso, 2017; Sliwinski et al., 2019; Patel et al., 2021; Karczmarz et al., 2021). While these works suggest that the Banzhaf value could be an alternative to the popular Shapley value-based model interpretation methods (Lundberg and Lee, 2017), it remains unclear in which settings the Banzhaf value may be preferable to the Shapley value. This work provides the first theoretical understanding of the advantage of the Banzhaf value in terms of robustness. In addition, the empirical study by Karczmarz et al. (2021) observes that the Banzhaf value is much more robust than the Shapley value when the numerical precision is low in the computation, which validates our theoretical result.

We would like to note that the Banzhaf value is a generalization of the Banzhaf power index (Banzhaf III, 1964) which is designed for gauging the voting power of players in a simple voting game. In a *simple voting game*, the utility function $U : 2^N \to \{0, 1\}$ where $U(\emptyset) = 0$, $U(N) = 1$ and $U(S) \leq U(T)$ whenever $S \subseteq T$. In contrast, the setting of data valuation is more complicated and challenging as we do *not* assume any particular structure of the utility function U. There are also many kinds of power indices available, such as Shapley-Shubik index (Shapley, 1953), Holler index (Holler, 1982), and Deegan-Packel index (Deegan and Packel, 1978). The interpretation and computation of these power indices are active topics in cooperative game theory (e.g., (Holler and Packel, 1983; Aziz, 2008)). In this work, we explore the most robust data value notion among the space of semivalues. Exploring the possibility of extending other kinds of cooperative solution concepts to data valuation is an interesting and promising future research direction.

Efficient Estimation of the Banzhaf and Shapley value. Most of the estimation algorithms for the Banzhaf and Shapley value are based on Monte Carlo techniques, especially when no prior knowledge about the structure of utility function U is available. The Simple Monte Carlo estimation for Shapley value (i.e., the Permutation Sampling) was mentioned in very early works (Mann and Shapley, 1960), and the sample complexity analysis of Permutation sampling for Shapley value can be found in (Maleki, 2015). Covert et al. propose an improved Shapley estimator based on the Importance Sampling technique. Jia et al. (2019b) improve the sample complexity of Monte Carlo-based Shapley estimation based on group testing technique (which is further improved by Wang and Jia (2023) later). G-Shapley, TMC-Shapley (Ghorbani and Zou, 2019) and KNN-Shapley (Jia et al., 2019a) have been proposed as the efficient proxies of Shapley value. However, these are *biased* estimators for the Shapley value in nature. The sample complexity of the Simple Monte Carlo method for the Banzhaf value / Banzhaf power index (Merrill III, 1982) first appeared in Bachrach et al. (2010).

Another line of works studies the estimation of the Shapley and Banzhaf value in the problems with specific structures, e.g., (weighted) voting games (Owen, 1972; Fatima et al., 2008; Teneggi et al., 2021), the games where only a few players have non-zero contribution (Jia et al., 2019b; Lin et al., 2022).

Alternative Approaches for Data Valuation in ML. We review some recent works on data valuation methods here that are *not* based on cooperative game theory, and we refer the readers to Sim et al. (2022) for a comprehensive technical survey of data valuation in ML. Sim et al. (2020) use the reduction in uncertainty of the model parameters given the data as the valuation metric. Axioms that are important for collaborative learning such as strict desirability and monotonicity are also mentioned in the paper. *Training-free* and *task-agnostic* data valuation methods have also been proposed. Tay et al. (2022) suggest a data valuation method utilizing maximum mean discrepancy (MMD) between the data source and the true data distribution. Xu et al. (2021) come up with a diversity measure called robust volume (RV) for valuing data sources. The robustness of the proposed data value notion is discussed in terms of the stability against data replication (via direct data copying). Han et al. (2020) also study the replication robustness of semivalues. Wu et al. (2022) use a domain-aware generalization bound for data valuation, where the bound is based on neural tangent kernel (NTK) theory. Amiri et al. (2022) use the statistical differences between the source data and a baseline dataset as the valuation metric.

B FURTHER DISCUSSION ABOUT CONSIDERATIONS IN DEFINITION 4.2

B.1 Why do we consider rank stability?

As mentioned in the main text, rank stability is a reasonable robustness measure as the ranking of data values is important in many applications such as data subset selection and data pruning. Yet, another natural robustness measure is the stability in absolute value. Specifically, we can view a semivalue as a function $\phi : \mathbb{R}^{2^n} \to \mathbb{R}^n$ which takes a utility function $U \in \mathbb{R}^{2^n}$ as input, and output the values of data points $\phi(U) \in \mathbb{R}^n$. By taking this functional view, a natural robustness measure for semivalue $\phi(\cdot; w)$ is its Lipschitz constant L, which is defined as the smallest constant such that

$$\left\|\phi(U; w) - \phi(\widehat{U}; w)\right\| \leq L \left\|U - \widehat{U}\right\| \tag{7}$$

for all possible pairs of U and $\widehat{U}$. However, since the efficiency axiom is relaxed for semivalue, such a robustness measure has the issue that *different semivalues have different scales; the same change in the absolute value may mean differently for them.* On the other hand, rank stability provides a fair measure for comparison between different semivalues.

B.2 Why do we consider a noise-structure-agnostic definition?

The safety margin defined in Definition 4.2 does **not** depend on the actual noise induced by a specific stochastic training algorithm. Indeed, one may attempt to **directly analyze the potential perturbation distribution of utility function** $U(S)$ **caused by the stochasticity in learning algorithm** (e.g., random initialization, mini-batch selection), and define the safety margin **with respect to the specific perturbation (distribution)**.

However, we did not adopt such a definition (referred to as *noise-structure-specific* notion later) due to several considerations.

I. Even if the datasets and learning algorithm are known in advance, it is usually intractable to analytically derive a definite assertion on the noise in performance scores. The probability distribution of the model performance scores $U(S) = \texttt{acc}(\mathcal{A}(S))$ change with different training data S, test data (characterized by $\texttt{acc}(\cdot)$), and the hyperparameters of learning algorithm $\mathcal{A}$. Even if such information is all available in advance (before we pick the data value notion), it is difficult to analyze the distribution of $U(S)$ without actually training on S for common learning algorithms such as SGD. In order to illustrate the technical difficulty, we show such an attempt in Section B.2.1.

Specifically, under the simple (if not the simplest) setting of 1-dimensional linear regression trained by full batch gradient descent with Gaussian random initialization, we show the derivation of the probability distribution of validation mean squared error.[7] We show that the distribution of validation mean squared error is a *generalized* χ^2 *distribution*. Unfortunately, the probability density and cumulative density of generalized χ^2 distribution are known for being intractable, which impedes further analysis of its impact on data values. Furthermore, *it is unclear how to analyze the validation loss distribution for batch stochastic gradient descent (see the discussion at the end of Appendix B.2.1).* As we can see, even for such a simple setting, there are significant challenges in designing noise-structure-specific robust data value notions.

II. Noise-structure-specific robust data value notion may be computationally infeasible. More importantly, in order to design noise-structure-specific robust data value notion for a dataset N of size n, we need to understand the noise distribution of $U(S)$ for every subset $S \subseteq N$, which introduces an exponentially large computationally burden. While one might be able to resort to numerical methods to approximate the intractable probability density issue mentioned before, the computational costs incurred by modeling the performance distribution on exponential subsets are prohibitive.

III. In practice, one may not have prior knowledge about the dataset and source of perturbation; in that case, a worst-case robustness notion is preferred. Another difficulty regarding the noise-structure-specific robustness notion is that in practice, the datasets may not be known in advance for privacy consideration (Agahari et al., 2022; Tian et al., 2022). Furthermore, the performance scores may also be perturbed due to other factors such as hardware faults, software bugs, or even adversarial attacks. Given the uncertainty of the perturbation in the practical scenario, it makes more sense to define the robustness in terms of the worst-case perturbation. This is exactly Kerckhoffs's principle and such definitions are common in machine learning. For instance, robust learning against adversarial examples is aimed at being resilient to the worst-case noise (Madry et al., 2017) within a norm ball; differential privacy (Dwork et al., 2006) protects individual data record's information from arbitrary attackers.

Based on the identified difficulties for the potential noise-structure-specific notion, and the advantages of the noise-structure-agnostic notion, we believe it is ideal to define the robustness through the way in Section 4.1. Importantly, the proposed

[7] We use full batch gradient descent so the only randomness in the learning algorithm is the random initialization.

robustness notion can not only lead to **tractable robustness analysis** for celebrated data value notions (Shapley, LOO); at the same time, the data value notion obtained by optimizing the proposed robustness measure, i.e., the Banzhaf value, indeed **achieves good robustness against realistic learning stochasticity** (as shown in the experiment section). Last but not least, the study of how to characterize the dependency between performance score and learning stochasticity itself needs to be investigated in depth before one could claim a reasonable probabilistic robustness definition under learning stochasticity.

B.2.1 Difficulties in Analytical Analysis of $U(S)$

As we mentioned in Section 3, the utility of a dataset $U(S)$ is defined as $\texttt{acc}(\mathcal{A}(S))$, i.e., the performance of a model $\mathcal{A}(S)$ trained on a dataset S. However, the learning algorithms may have randomness during the training process. For instance, the stochastic gradient descent (SGD) algorithm involves random weights initialization and random mini-batch selection. Formally, we can view $\mathcal{A}$ as a randomized function that takes a dataset S as input, and output a trained model $\mathcal{A}(S; r)$, where r is a random string that describes all randomness used during the training process. For SGD, r is the weights initialization and mini-batch selection choices. For each execution of the learning algorithm, r is sampled from the corresponding probability distribution $\mathcal{P}_{\mathcal{A}}$ specified by the learning algorithm, e.g., the weights are initialized by isotropic Gaussian distribution and mini-batch selection is based on Binomial distribution. Since the trained model $\mathcal{A}(S)$ is a random variable, the utility $U(S)$ is inherently a randomized function and the randomness depends on r. To make data value notions such as Shapley and Banzhaf value to be well-defined and independent from the learning stochasticity, at the end of Section 3 we refine $U(S) := \mathbb{E}_{r\sim P}[\texttt{acc}(\mathcal{A}(S; r))]$. Let $\widehat{U}(S)$ denote the random utility $\texttt{acc}(\mathcal{A}(S; r))$ with a randomly $r \sim \mathcal{P}$. In order to find the most robust semivalue against the random noise $\widehat{U}(S) - U(S)$, one natural idea would be analytically derive the probability distribution of $\widehat{U}(S)$, and design the corresponding data value accordingly.

Unfortunately, it is actually non-trivial to conclude a definite assumption on the performance noise caused by learning stochasticity. In this section, we illustrate such difficulty by directly analyzing the distribution of $\widehat{U}(S)$ for arguably the simplest setting: 1-dimensional linear regression trained by gradient descent with random initialization. The model here is defined as $f(x; \theta) := \theta x$ and is trained on a dataset $S = \{(x, y)\}$ via mean squared error $\mathcal{L}(\theta; S) = \frac{1}{2}\sum_{(x,y)\in S}(y - f(x; \theta))^2$. The space of both input feature x and model parameter θ are $\mathbb{R}$.

The source of the randomness in the learning algorithm here is random initialization. Specifically, we use gradient descent to train the linear regression model with Gaussian random initialization.

$$\begin{aligned}
&\theta_0 \sim \mathcal{N}(0, \sigma^2)\\
&\theta_1 \leftarrow \theta_0 - \eta\nabla\mathcal{L}(\theta_0)\\
&\quad\cdots\\
&\theta_T \leftarrow \theta_0 - \eta\nabla\mathcal{L}(\theta_{T-1})
\end{aligned}$$

where η is the learning rate and T is the total number of iterations. The utility of the trained model is given by the mean squared error $\widehat{U}(S) := \mathcal{L}(\theta_T; S_{\text{val}})$ on validation set $S_{\text{val}} = \{(x^*, y^*)\}$.

Due to the simplicity of the setting, we can derive the evolution of the distribution of θ_T as well as $\mathcal{L}(\theta_T; S_{\text{val}})$. Since $\nabla\mathcal{L}(\theta) = \sum_{(x,y)\in S} x(x\theta - y)$, at iteration t we have

$$\begin{aligned}
\theta_{t+1} &= \theta_t - \eta\sum_{(x,y)\in S} x(x\theta_t - y)\\
&= \theta_t\left(1 - \eta\sum_{(x,y)\in S} x^2\right) + \eta\sum_{(x,y)\in S} xy
\end{aligned}$$

Therefore, $\{\theta_t\}$ is a sequence of Gaussian random variables, where

$$\begin{aligned}
\mathbb{E}[\theta_{t+1}] &= \mathbb{E}[\theta_t]\left(1 - \eta\sum_{(x,y)\in S} x^2\right) + \eta\sum_{(x,y)\in S} xy\\
\text{Var}(\theta_{t+1}) &= \text{Var}(\theta_t)\left(1 - \eta\sum_{(x,y)\in S} x^2\right)^2
\end{aligned}$$

To simplify the notation, let $f(S) = \sum_{(x,y)\in S} xy$ and $g(S) = \sum_{(x,y)\in S} x^2$. By a simple analysis, we can derive the general term formula for θ_T as

$$\mathbb{E}[\theta_T] = \frac{f(S)}{g(S)}\left(1 - (1-\eta g(S))^T\right)$$
$$\mathrm{Var}(\theta_T) = \sigma^2\left(1-\eta g(S)\right)^{2T}$$

For each validation data point (x^*, y^*), we have

$$x^*\theta_T - y^* \sim \mathcal{N}\left(x^*\mathbb{E}[\theta_T] - y^*, \mathrm{Var}(\theta_T)(x^*)^2\right)$$

Therefore, $(x^*\theta_T - y^*)^2$ is a (scaled) non-central χ'^2 distribution with degree of freedom 1 and non-centrality parameter $\frac{(x^*\mathbb{E}[\theta_T]-y^*)^2}{\mathrm{Var}(\theta_T)(x^*)^2}$. (Abramowitz and Stegun (1964), Section 26.4.25):

$$\begin{aligned}(x^*\theta_T - y^*)^2 &\sim \mathrm{Var}(\theta_T)(x^*)^2 \cdot \chi'^2\left(1, \left(\frac{x^*\mathbb{E}[\theta_T]-y^*}{\texttt{Std}(\theta_T)(x^*)}\right)^2\right)\\ &\sim \mathrm{Var}(\theta_T)(x^*)^2 \cdot \chi'^2\left(1, \frac{\left(x^*\mathbb{E}[\theta_T]-y^*\right)^2}{\mathrm{Var}(\theta_T)(x^*)^2}\right)\end{aligned}$$

Consequently, the validation loss $\mathcal{L}(\theta_T; S_{\mathrm{val}}) = \sum_{(x^*,y^*)\in S_{\mathrm{val}}}(x^*\theta_T - y^*)^2$ is a generalized chi-squared distribution (Davies, 1980):

$$\mathcal{L}(\theta_T; S_{\mathrm{val}}) \sim \sum_{(x^*,y^*)\in S_{\mathrm{val}}} \mathrm{Var}(\theta_T)(x^*)^2 \cdot \chi'^2\left(1, \frac{\left(x^*\mathbb{E}[\theta_T]-y^*\right)^2}{\mathrm{Var}(\theta_T)(x^*)^2}\right)$$

There are two major difficulties in applying the above results for designing robust data value notions specific to such a simplified setting:

1. The generalized chi-squared distribution is known for intractable probability density or cumulative distribution (Davies, 1980; Das and Geisler, 2021).
2. To design noise-structure-specific robust data value notion for a dataset N of size n, we need to understand the noise distribution for every subset $S \subseteq N$, which introduces an exponentially large computational burden. While one might be able to resort to numerical methods to approximate the intractable probability density, the computational costs incurred by modeling the performance distribution on exponential subsets are prohibitive.

Therefore, even for such a very simple setting, there are significant difficulties in designing noise-structure-specific robust data value notions based on directly analyzing noise distribution.

Difficulties in Extending to Batch Stochastic Gradient Descent. Furthermore, it is unclear how to extend the above analysis to batch stochastic gradient descent. For the case of batch stochastic gradient descent, while it is easy to see that the parameter in the first iteration θ_1 is a Gaussian mixture, the distribution of parameters after the first iteration is intractable to analyze.

Based on the above-identified difficulties, it is preferable to define the robustness in a noise-structure-agnostic way as we did in Section 4.

B.3 Why do we consider a U-structure-agnostic definition?

This consideration shares the same reason as **II** and **III** in Appendix B.2. If the safety margin depends on the specific U, it means that we need to know about $U(S)$ for every $S \subseteq N$, which is computationally infeasible or even impossible. On the other hand, the definition of "τ-distinguishable" utility functions characterizes the collection of Us such that the semivalue should be robust on, while also leading to tractable robustness analysis for semivalues.

C PROOFS AND ADDITIONAL THEORETICAL RESULTS

We provide a summary of the content in this section for the convenience of the readers.

- Appendix C.1: Proofs for the theorems appeared in the maintext.
- Appendix C.2: The MSR Estimator does not Extend to the Shapley Value and Other Known Semivalues.
- Appendix C.3: Robustness of the MSR Estimator.
- Appendix C.4: Stability of Banzhaf value in ℓ_2-norm.

C.1 Proofs for Theorem 4.4, 4.6, 4.8, 4.9, 4.10 in the Main Text

We omit the parameters of U, N, or w when it's clear from the context.

C.1.1 The Safety Margin for the LOO error, the Shapley value, and the Banzhaf value

Lemma C.1. *Given a semivalue with weight function $w(\cdot)$, we have*

$$\texttt{Safe}(\tau; w) = \tau \sqrt{\frac{\left(\sum_{k=1}^{n-1} \binom{n-2}{k-1} \left(w(k) + w(k+1)\right)\right)^2}{\sum_{k=1}^{n-1} \binom{n-2}{k-1} \left(w(k) + w(k+1)\right)^2}} \tag{8}$$

for any $\tau > 0$.

Proof. For any $\tau > 0$ and any pair of (i, j), we denote

$$\texttt{Safe}_{i,j}(\tau; w) = \min_{U \in \mathcal{U}_{i,j}^{(\tau)}} \min_{\widehat{U} \in \{\widehat{U} : D_{i,j}(U;w) D_{i,j}(\widehat{U};w) \le 0\}} \left\| \widehat{U} - U \right\|$$

as the minimum amount of noise that is required to reverse the ranking of (i, j) among all utility functions that τ-distinguish (i, j). Thus, the safety margin of the semivalue w is

$$\texttt{Safe}(\tau; w) = \min_{i \neq j} \texttt{Safe}_{i,j}(\tau; w)$$

Note that $D_{i,j}(U; w)$ can be written as a dot product of U and a column vector $a \in \mathbb{R}^{2^n}$

$$D_{i,j}(U; w) = a^T U$$

where each entry of a corresponds to a subset $S \subseteq N$. We use $a[\cdot]$ to denote the value of a's entry corresponds to S. For all $S \subseteq N \setminus \{i, j\}$, $a[S \cup i] = w(|S| + 1) + w(|S| + 2)$ and $a[S \cup j] = -(w(|S| + 1) + w(|S| + 2))$, and for all other subsets $a[S] = 0$. Let the perturbation $x = \widehat{U} - U$ and matrix $A = aa^T$.

$$\begin{aligned} D_{i,j}(U; w) D_{i,j}(\widehat{U}; w) &= (a^T U)(a^T \widehat{U}) \\ &= (a^T U)^T (a^T \widehat{U}) \\ &= U^T a a^T \widehat{U} \\ &= U^T A \widehat{U} \\ &= U^T A (U + x) \end{aligned}$$

Thus, if $D_{i,j}(U;w)D_{i,j}(\widehat{U};w) \leq 0$, the size of the perturbation x must be at least

$$\begin{aligned}\|x\| &\geq \frac{|U^T AU|}{\|U^T A\|} \\ &= \frac{|U^T AU|}{\sqrt{U^T AAU}} \\ &= \frac{|U^T AU|}{\sqrt{a^T a}\sqrt{|U^T AU|}} \\ &= \sqrt{\frac{|U^T AU|}{a^T a}}\end{aligned} \tag{9}$$

where (9) is because $AA = (aa^T)(aa^T) = a(a^T a)a^T = (a^T a)aa^T = (a^T a)A$. This lower bound is achievable when we set x on the direction of $U^T A$. Therefore, we have

$$\texttt{Safe}_{i,j}(\tau; w) = \min_{U \in \mathcal{U}_{i,j}^{(\tau)}} \sqrt{\frac{|U^T AU|}{a^T a}}$$

To make the notations less cumbersome, denote $f(S) = w(|S|+1) + w(|S|+2)$, and $g(S) = U(S \cup i) - U(S \cup j)$. By expanding the expression, we have

$$\begin{aligned}\frac{|U^T AU|}{a^T a} &= \frac{\left|\sum_{S_1 \subseteq N\setminus\{i,j\}} \sum_{S_2 \subseteq N\setminus\{i,j\}} f(S_1)f(S_2)g(S_1)g(S_2)\right|}{\sum_{S \subseteq N\setminus\{i,j\}} f^2(S)} \\ &= \frac{\left|\left(\sum_{S_1 \subseteq N\setminus\{i,j\}} f(S_1)g(S_1)\right)\left(\sum_{S_2 \subseteq N\setminus\{i,j\}} f(S_2)g(S_2)\right)\right|}{\sum_{S \subseteq N\setminus\{i,j\}} f^2(S)} \\ &= \frac{\left(\sum_{S \subseteq N\setminus\{i,j\}} f(S)g(S)\right)^2}{\sum_{S \subseteq N\setminus\{i,j\}} f^2(S)} \\ &= \frac{\left(\sum_{k=1}^{n-1} (w(k)+w(k+1)) \sum_{S \subseteq N\setminus\{i,j\}, |S|=k-1} (U(S\cup i) - U(S \cup j))\right)^2}{\sum_{k=1}^{n-1} \binom{n-2}{k-1} (w(k)+w(k+1))^2} \\ &= \frac{\left(\sum_{k=1}^{n-1} \binom{n-2}{k-1} (w(k)+w(k+1)) \binom{n-2}{k-1}^{-1} \sum_{S \subseteq N\setminus\{i,j\}, |S|=k-1} (U(S\cup i) - U(S \cup j))\right)^2}{\sum_{k=1}^{n-1} \binom{n-2}{k-1} (w(k)+w(k+1))^2}\end{aligned} \tag{10}$$

Clearly, the minimum of (10) is achieved when

$$\binom{n-2}{k-1}^{-1} \sum_{S \subseteq N\setminus\{i,j\}, |S|=k-1} (U(S\cup i) - U(S \cup j)) = \tau$$

for all k, i.e., $\texttt{Safe}_{i,j}(\tau; w) = \tau\sqrt{\frac{\left(\sum_{k=1}^{n-1} \binom{n-2}{k-1}(w(k)+w(k+1))\right)^2}{\sum_{k=1}^{n-1} \binom{n-2}{k-1}(w(k)+w(k+1))^2}}$. This holds for every pair of data points (i,j), which leads to our conclusion where $\texttt{Safe}(\tau; w) = \tau\sqrt{\frac{\left(\sum_{k=1}^{n-1} \binom{n-2}{k-1}(w(k)+w(k+1))\right)^2}{\sum_{k=1}^{n-1} \binom{n-2}{k-1}(w(k)+w(k+1))^2}}$. □

The Safety Margin for the LOO error and the Shapley value.

Theorem 4.4 (restated). For any $\tau > 0$, Leave-one-out error ($w_{\text{loo}}(k) = n\mathbf{1}[k=n]$) achieves $\texttt{Safe}(\tau; w_{\text{loo}}) = \tau$, and Shapley value ($w_{\text{shap}}(k) = \binom{n-1}{k-1}^{-1}$) achieves $\texttt{Safe}(\tau; w_{\text{shap}}) = \tau\frac{n-1}{\sqrt{\sum_{k=1}^{n-1}\binom{n-2}{k-1}^{-1}}}$.

Proof. By plugging in $w_{\text{loo}}(k) = n\mathbf{1}[k=n]$ to (8), we have $\texttt{Safe}(\tau; w_{\text{loo}}) = \tau$. By plugging in $w_{\text{shap}}(k) = \binom{n-1}{k-1}^{-1}$ to (8), we have $\texttt{Safe}(\tau; w_{\text{shap}}) = \tau\frac{n-1}{\sqrt{\sum_{k=1}^{n-1}\binom{n-2}{k-1}^{-1}}}$. □

Banzhaf Value Achieves the Largest Safety Margin.

Theorem 4.6 (restated). For any $\tau > 0$, the Banzhaf value ($w(k) = \frac{n}{2^{n-1}}$) achieves the largest safety margin $\texttt{Safe}(\tau; w) = \tau 2^{n/2-1}$ among all semivalues.

Proof. By Lemma C.1, we want to find the optimal semivalue weight function w that maximizes $\frac{\left(\sum_{k=1}^{n-1}\binom{n-2}{k-1}(w(k)+w(k+1))\right)^2}{\sum_{k=1}^{n-1}\binom{n-2}{k-1}(w(k)+w(k+1))^2}$. Notice that

$$\begin{aligned}
\frac{\left(\sum_{k=1}^{n-1}\binom{n-2}{k-1}\left(w(k)+w(k+1)\right)\right)^2}{\sum_{k=1}^{n-1}\binom{n-2}{k-1}\left(w(k)+w(k+1)\right)^2} &= \frac{\left(\sum_{k=1}^{n-1}\sqrt{\binom{n-2}{k-1}}\sqrt{\binom{n-2}{k-1}}\left(w(k)+w(k+1)\right)\right)^2}{\sum_{k=1}^{n-1}\binom{n-2}{k-1}\left(w(k)+w(k+1)\right)^2} \\
&\leq \frac{\sum_{k=1}^{n-1}\binom{n-2}{k-1}\sum_{k=1}^{n-1}\binom{n-2}{k-1}\left(w(k)+w(k+1)\right)^2}{\sum_{k=1}^{n-1}\binom{n-2}{k-1}\left(w(k)+w(k+1)\right)^2} \qquad (11) \\
&= \sum_{k=1}^{n-1}\binom{n-2}{k-1} \\
&= 2^{n-2}
\end{aligned}$$

where (11) is due to Cauchy-Schwarz inequality.

Note that this upper bound is achievable whenever $w(|S|+1) + w(|S|+2)$ is a constant due to the equality condition of Cauchy-Schwarz, which the weight function of Banzhaf value clearly satisfies. Therefore, the Banzhaf value achieves the largest $\texttt{Safe}(\tau; w)$ among all possible semivalues. □

Further Discussion of Theorem 4.6. The safety margin is the largest noise in utility that can be tolerated such that the ranking of *exact* semivalues calculated from clean utility match with that of *exact* semivalues calculated from a noisy utility. However, calculating exact semivalues for a given utility function is NP-hard in general, and in practice, one often resorts to evaluating the utility function at limited sampled subsets and then using these limited samples to approximate semivalues. Hence, a natural question to ask is whether we can characterize the maximally-tolerable utility noise on the *limited* sampled subsets such that the ranking of *approximate* semivalues calculated from the clean utility samples align with that of *approximate* semivalues calculated from the noisy samples. However, one issue with this type of characterization is that the "safety margin" in this case depends on *both* the expression of the semivalue (i.e., w that parameterizes the semivalue), as well as the underlying estimation algorithm for that semivalue. Since different semivalues have different estimation algorithms, such a result for different semivalues is not really comparable. On the other hand, our result in Theorem 4.6 lifts the dependence on the underlying estimation algorithm. As a consequence, it allows one to compare the robustness between different semivalues.

We also note that, the Banzhaf value is *not* the unique semivalue that achieves the maximal robustness in the setup of Theorem 4.6. Any semivalues with a weight function w s.t. $w(k) + w(k+1)$ is a constant also achieve the same safety margin. Such a semivalue must have $w(1) = w(3) = w(5) = \ldots$ and $w(2) = w(4) = w(6) = \ldots$. However, there's no natural explanation for why the semivalue should weigh odd and even cardinalities differently. Hence, the Banzhaf value is the only "reasonable" semivalue with maximal robustness.

C.1.2 Sample Complexity of Simple MC and MSR Estimator.

Theorem 4.8 (restated). $\widehat{\phi}_{\mathrm{MC}}$ is an (ε, δ)-approximation to the exact Banzhaf value in ℓ_2-norm with $O(\frac{n^2}{\varepsilon^2}\log(\frac{n}{\delta}))$ calls of $U(\cdot)$, and in ℓ_∞-norm with $O(\frac{n}{\varepsilon^2}\log(\frac{n}{\delta}))$ calls of $U(\cdot)$.

Proof. Let $\mathcal{S} = \{S_1, \ldots, S_m\}$ be the samples used for computing $\widehat{\phi}_{\mathrm{MC}}(i)$. Since the marginal contribution $U(S \cup i) - U(S)$ is always bounded between $[-1, 1]$, by Hoeffding, we have

$$\Pr\left[\left|\widehat{\phi}_{\mathrm{MC}}(i) - \phi(i)\right| \geq \varepsilon\right] \leq 2\exp\left(-2m\varepsilon^2\right)$$

which holds for every $i \in N$.

Thus, with union bound, for ℓ_2-norm we have

$$\begin{aligned}
\Pr_{\widehat{\phi}_{\mathrm{MC}}}\left[\left\|\widehat{\phi}_{\mathrm{MC}}-\phi\right\|_2\geq\varepsilon\right] &= 1-\Pr_{\widehat{\phi}_{\mathrm{MC}}}\left[\left\|\widehat{\phi}_{\mathrm{MC}}-\phi\right\|_2^2\leq\varepsilon^2\right]\\
&\leq 1-\Pr_{\widehat{\phi}_{\mathrm{MC}}}\left[\cap_i\left|\widehat{\phi}_{\mathrm{MC}}(i)-\phi(i)\right|\leq\varepsilon/\sqrt{n}\right]\\
&= \Pr_{\widehat{\phi}_{\mathrm{MC}}}\left[\cup_i\left|\widehat{\phi}_{\mathrm{MC}}(i)-\phi(i)\right|\geq\varepsilon/\sqrt{n}\right]\\
&\leq \sum_{i=1}^{n}\Pr_{\widehat{\phi}_{\mathrm{MC}}}\left[\left|\widehat{\phi}_{\mathrm{MC}}(i)-\phi(i)\right|\geq\varepsilon/\sqrt{n}\right]\\
&\leq 2n\exp\left(-2m\varepsilon^2/n\right)
\end{aligned}$$

By setting $2n\exp\left(-2m\varepsilon^2/n\right)\leq\delta$, we get $m\geq\frac{n}{2\varepsilon^2}\log\left(\frac{2n}{\delta}\right)=O\left(\frac{n}{\varepsilon^2}\log(\frac{n}{\delta})\right)$. However, this m only corresponds to the number of samples used to estimate a single $\phi(i)$, so the total number of samples required is $O\left(\frac{n^2}{\varepsilon^2}\log(\frac{n}{\delta})\right)$.

For ℓ_∞-norm we have

$$\begin{aligned}
\Pr_{\widehat{\phi}_{\mathrm{MC}}}\left[\left\|\widehat{\phi}_{\mathrm{MC}}-\phi\right\|_\infty\geq\varepsilon\right] &= \Pr_{\widehat{\phi}_{\mathrm{MC}}}\left[\cup_i\left|\widehat{\phi}_{\mathrm{MC}}(i)-\phi(i)\right|\geq\varepsilon\right]\\
&\leq \sum_{i=1}^{n}\Pr_{\widehat{\phi}_{\mathrm{MC}}}\left[\left|\widehat{\phi}_{\mathrm{MC}}(i)-\phi(i)\right|\geq\varepsilon\right]\\
&\leq 2n\exp\left(-2m\varepsilon^2\right)
\end{aligned}$$

By setting $2n\exp\left(-2m\varepsilon^2\right)\leq\delta$, we get $m\geq\frac{1}{2\varepsilon^2}\log\left(\frac{2n}{\delta}\right)=O\left(\frac{1}{\varepsilon^2}\log(\frac{n}{\delta})\right)$. However, this m only corresponds to the number of samples used to estimate a single $\phi(i)$, so the total number of samples required is $O\left(\frac{n}{\varepsilon^2}\log(\frac{n}{\delta})\right)$. □

Theorem 4.9 (restated). $\widehat{\phi}_{\mathrm{MSR}}$ is an (ε,δ)-approximation to the exact Banzhaf value in ℓ_2-norm with $O\left(\frac{n}{\varepsilon^2}\log(\frac{n}{\delta})\right)$ calls of $U(\cdot)$, and in ℓ_∞-norm with $O\left(\frac{1}{\varepsilon^2}\log(\frac{n}{\delta})\right)$ calls of $U(\cdot)$.

Proof. Since $\mathcal{S}=\{S_1,\ldots,S_m\}$ each i.i.d. drawn from $\mathrm{Unif}(2^N)$, it is easy to see that the size of sampled subsets that include data point i follows binomial distribution $|\mathcal{S}_{\ni i}|\sim\mathrm{Bin}(m,0.5)$, and $|\mathcal{S}_{\ni i}|=m-|\mathcal{S}_{\not\ni i}|$.

We first define an alternative estimator

$$\widetilde{\phi}(i)=\frac{1}{m/2}\sum_{S\in\mathcal{S}_{\ni i}}U(S)-\frac{1}{m/2}\sum_{S\in\mathcal{S}_{\not\ni i}}U(S)$$

which is independent of $|\mathcal{S}_{\ni i}|$ and $|\mathcal{S}_{\not\ni i}|$. When both $|\mathcal{S}_{\ni i}|$ and $|\mathcal{S}_{\not\ni i}|>0$, we have

$$\begin{aligned}
\left|\widehat{\phi}(i)-\widetilde{\phi}(i)\right| &= \left|\left(\frac{1}{|\mathcal{S}_{\ni i}|}-\frac{1}{m/2}\right)\sum_{S\in\mathcal{S}_{\ni i}}U(S)-\left(\frac{1}{|\mathcal{S}_{\not\ni i}|}-\frac{1}{m/2}\right)\sum_{S\in\mathcal{S}_{\not\ni i}}U(S)\right|\\
&\leq \left|1-\frac{2|\mathcal{S}_{\ni i}|}{m}\right|+\left|1-\frac{2|\mathcal{S}_{\not\ni i}|}{m}\right| \qquad (12)\\
&= 2\left|1-\frac{2|\mathcal{S}_{\ni i}|}{m}\right| \qquad (13)\\
&= \frac{4}{m}\left||\mathcal{S}_{\ni i}|-\frac{m}{2}\right|
\end{aligned}$$

where (12) is due to $U(S)\leq 1$ and (13) is due to $|\mathcal{S}_{\ni i}|=m-|\mathcal{S}_{\not\ni i}|$. When one of $|\mathcal{S}_{\ni i}|$ and $|\mathcal{S}_{\not\ni i}|=0$, this upper bound also clearly holds.

Since $|\mathcal{S}_{\ni i}|\sim\mathrm{Bin}(m,0.5)$, by Hoeffding inequality we have

$$\Pr\left[\left||\mathcal{S}_{\ni i}|-\frac{m}{2}\right|\geq\Delta\right]\leq 2\exp\left(-\frac{2\Delta^2}{m}\right)$$

Hence, with probability at least $1 - 2\exp\left(-\frac{2\Delta^2}{m}\right)$, we have

$$\left|\widehat{\phi}(i) - \widetilde{\phi}(i)\right| \leq \frac{4\Delta}{m}$$

Since

$$\begin{aligned}\widetilde{\phi}(i) &= \frac{2}{m}\left(\sum_{S\in\mathcal{S}_{\ni i}} U(S) - \sum_{S\in\mathcal{S}_{\not\ni i}} U(S)\right) \\ &= \frac{1}{m}\sum_{S\in\mathcal{S}} 2U(S)\text{sign}(i,S)\end{aligned}$$

where $\text{sign}(i,S) = 2\mathbf{1}[i\in S] - 1 \in \{\pm 1\}$. Thus, $2U(S)\text{sign}(i,S) \in [-2,2]$ and we can apply Hoeffding to bound the tail of $|\widetilde{\phi}(i) - \phi(i)|$:

$$\Pr\left[|\widetilde{\phi}(i) - \phi(i)| \geq t\right] \leq 2\exp\left(-\frac{mt^2}{8}\right)$$

Now we bound $|\widehat{\phi}(i) - \phi(i)|$ as follows:

$$\begin{aligned}&\Pr\left[|\widehat{\phi}(i) - \phi(i)| \geq \varepsilon\right] \\ &= \Pr\left[|\widehat{\phi}(i) - \phi(i)| \geq \varepsilon \middle| \left||\mathcal{S}_{\ni i}| - \frac{m}{2}\right| \leq \Delta\right] \Pr\left[\left||\mathcal{S}_{\ni i}| - \frac{m}{2}\right| \leq \Delta\right] \\ &\quad + \Pr\left[|\widehat{\phi}(i) - \phi(i)| \geq \varepsilon \middle| \left||\mathcal{S}_{\ni i}| - \frac{m}{2}\right| > \Delta\right] \Pr\left[\left||\mathcal{S}_{\ni i}| - \frac{m}{2}\right| > \Delta\right] \\ &\leq \Pr\left[|\widehat{\phi}(i) - \phi(i)| \geq \varepsilon \middle| \left||\mathcal{S}_{\ni i}| - \frac{m}{2}\right| \leq \Delta\right] + 2\exp\left(-\frac{2\Delta^2}{m}\right) \\ &\leq \Pr\left[|\widetilde{\phi}(i) - \phi(i)| \geq \varepsilon - \frac{4\Delta}{m} \middle| \left||\mathcal{S}_{\ni i}| - \frac{m}{2}\right| \leq \Delta\right] + 2\exp\left(-\frac{2\Delta^2}{m}\right) \\ &\leq \frac{\Pr\left[|\widetilde{\phi}(i) - \phi(i)| \geq \varepsilon - \frac{4\Delta}{m}\right]}{1 - 2\exp\left(-\frac{2\Delta^2}{m}\right)} + 2\exp\left(-\frac{2\Delta^2}{m}\right) \\ &\leq \frac{2\exp\left(-\frac{1}{8}m(\varepsilon - \frac{4\Delta}{m})^2\right)}{1 - 2\exp\left(-\frac{2\Delta^2}{m}\right)} + 2\exp\left(-\frac{2\Delta^2}{m}\right) \\ &\leq 3\exp\left(-\frac{1}{8}m\left(\varepsilon - \frac{4\Delta}{m}\right)^2\right) + 2\exp\left(-\frac{2\Delta^2}{m}\right)\end{aligned}$$

where the last inequality holds whenever $1 - 2\exp\left(-\frac{2\Delta^2}{m}\right) \geq \frac{2}{3}$.

We can then optimize this bound by setting $-\frac{1}{8}m\left(\varepsilon - \frac{4\Delta}{m}\right)^2 = -\frac{2\Delta^2}{m}$, where we obtain $\Delta = \frac{m\varepsilon}{8}$, and the bound becomes

$$\begin{aligned}\Pr\left[|\widehat{\phi}(i) - \phi(i)| \geq \varepsilon\right] &\leq 3\exp\left(-\frac{1}{8}m\left(\varepsilon - \frac{4\Delta}{m}\right)^2\right) + 2\exp\left(-\frac{2\Delta^2}{m}\right) \\ &= 5\exp\left(-\frac{m\varepsilon^2}{32}\right)\end{aligned}$$

By union bound, we have

$$\Pr_{\widehat{\phi}_{\text{MSR}}}\left[\left\|\widehat{\phi}_{\text{MSR}} - \phi\right\|_2 \geq \varepsilon\right] \leq 5n\exp\left(-\frac{m\varepsilon^2}{32n}\right)$$

and

$$\Pr_{\widehat{\phi}_{\mathrm{MSR}}}\left[\left\|\widehat{\phi}_{\mathrm{MSR}}-\phi\right\|_\infty \geq \varepsilon\right] \leq 5n\exp\left(-\frac{m\varepsilon^2}{32}\right)$$

By setting $\delta \leq 5n\exp\left(-\frac{m\varepsilon^2}{32n}\right)$, we obtain the sample complexity $O\left(\frac{n}{\varepsilon^2}\log(\frac{n}{\delta})\right)$ for ℓ_2 norm, and by setting $\delta \leq 5n\exp\left(-\frac{m\varepsilon^2}{32}\right)$, we obtain the sample complexity $O\left(\frac{1}{\varepsilon^2}\log(\frac{n}{\delta})\right)$ for ℓ_∞-norm. □

C.1.3 Lower Bound for the Banzhaf Value Estimator

Theorem 4.10 (restated). Every (possibly randomized) Banzhaf value estimation algorithm that achieves (ε,δ)-approximation in ℓ_∞-norm for constant $\delta \in (0,1/2)$ has sample complexity at least $\Omega(\frac{1}{\varepsilon})$.

Proof. To show the lower bound of the sample complexity for Banzhaf value estimation, we use *Yao's minimax principle*: to show a lower bound on a randomized algorithm, it suffices to define a distribution on some family of instances and show a lower bound for deterministic algorithms on this distribution.

Fix $\varepsilon \in (0,1)$. We define the collection of instance $\mathcal{I}_0$ as all utility functions U such that $U(S) = U(S\cup n)$ for all $S \subseteq [n-1]$. We define the collection of instance $\mathcal{I}_1$ as all utility functions U s.t. there are exactly $2^{n-1}(2\varepsilon)$ of the $S \subseteq [n-1]$ has $U(S)=0, U(S\cup n)=1$, and for all other S we have $U(S)=U(S\cup n)$. We define a distribution over $\mathcal{I}_0\cup\mathcal{I}_1$ by first randomly picking $\mathcal{I}_0$ or $\mathcal{I}_1$ with probability 1/2, and then picking a utility function from the selected instance class uniformly at random.

For any $U \in \mathcal{I}_0$, we have $\phi(n;U)=0$, and for any $U\in\mathcal{I}_1$, we have $\phi(n;U)=\frac{2^{n-1}(2\varepsilon)}{2^{n-1}}=2\varepsilon$. Thus, in order to achieve $\left\|\widehat{\phi}-\phi\right\|_\infty < \varepsilon$, the estimator must be able to distinguish between whether the utility function is from $\mathcal{I}_0$ or $\mathcal{I}_1$. For this, it needs to identify at least one $S\subseteq[n-1]$ s.t. $U(S)=0, U(S\cup n)=1$. However, since those S are chosen uniformly at random, no matter what sampling strategy the algorithm has, each query succeeds with probability at most $2^{n-1}(2\varepsilon)/2^{n-1}=2\varepsilon$. Thus, for m queries, the total failure probability is at least $(1-2m\varepsilon)/2$. To make the failure probability at most δ, we need number of samples m s.t. $(1-2m\varepsilon)/2\leq\delta$, which leads to the lower bound $m\geq\frac{1-2\delta}{2\varepsilon}$. Thus, we have $m\in\Omega(\frac{1}{\varepsilon})$. □

C.2 MSR Estimator does not Extend to the Shapley Value and Other Known Semivalues

In this section, we provide proof and more discussion about why the existence of the MSR estimator is a unique advantage of Banzhaf value.

Numerical Instability. It is easy to see that semivalue can be written as the expectation of *weighted* marginal contribution

$$\begin{aligned}
\phi_{\text{semi}}\left(i; U, N, w\right) &:= \frac{1}{n}\sum_{k=1}^{n} w(k) \sum_{S\subseteq N\setminus i, |S|=k-1} \left[U(S\cup i) - U(S)\right] \\
&= \mathbb{E}_{S\sim \text{Unif}(2^{N\setminus i})}\left[\frac{2^{n-1}w(|S|+1)}{n}\left(U(S\cup i) - U(S)\right)\right] \\
&= \mathbb{E}_{S\sim \text{Unif}(2^{N\setminus i})}\left[\frac{2^{n-1}w(|S|+1)}{n}U(S\cup i)\right] - \mathbb{E}_{S\sim \text{Unif}(2^{N\setminus i})}\left[\frac{2^{n-1}w(|S|+1)}{n}U(S)\right] \\
&= \mathbb{E}_{S\sim \text{Unif}(\{S\in 2^N : S\ni i\})}\left[\frac{2^{n-1}w(|S|)}{n}U(S)\right] - \mathbb{E}_{S\sim \text{Unif}(2^{N\setminus i})}\left[\frac{2^{n-1}w(|S|+1)}{n}U(S)\right] \qquad (14)
\end{aligned}$$

Hence, a straightforward way to design MSR estimator for arbitrary semivalue is to to sample $\mathcal{S} = \{S_1, \ldots, S_m\}$ each i.i.d. drawn from $\text{Unif}(2^N)$, and estimate $\phi(i)$ as

$$\widehat{\phi}_{\text{MSR}}(i) = \frac{1}{|\mathcal{S}_{\ni i}|}\sum_{S\in\mathcal{S}_{\ni i}} \frac{2^{n-1}w(|S|)}{n}U(S) - \frac{1}{|\mathcal{S}_{\not\ni i}|}\sum_{S\in\mathcal{S}_{\not\ni i}} \frac{2^{n-1}w(|S|+1)}{n}U(S)$$

For Shapley value, $w(|S|) = \binom{n-1}{|S|-1}^{-1}$, which makes the MSR estimator for Shapley value becomes

$$\widehat{\phi}_{\text{MSR}}(i) = \frac{1}{|\mathcal{S}_{\ni i}|}\sum_{S\in\mathcal{S}_{\ni i}} \frac{1}{n}\binom{n-1}{|S|-1}^{-1}U(S) - \frac{1}{|\mathcal{S}_{\not\ni i}|}\sum_{S\in\mathcal{S}_{\not\ni i}} \frac{1}{n}\binom{n-1}{|S|}^{-1}U(S)$$

which is numerically unstable for large n due to the combinatorial coefficients.

Impossible to construct a special sampling distribution for MSR. The MSR estimator for Banzhaf value samples from $\text{Unif}(2^N)$ since for a random set $S\sim\text{Unif}(2^N)$, we have its conditional distribution $S|S\not\ni i \sim \text{Unif}(2^{N\setminus i})$ and $S|S\ni i \sim \text{Unif}(\{S\in 2^N : S\ni i\})$, which exactly matches the two distributions the expectation in (14) is taken over for Banzhaf value. For a semivalue with weight function w, can we design a similar distribution $\mathcal{D}$ over 2^N so that $\phi_{\text{semi}}\left(i; U, N, w\right) = \mathbb{E}_{S\sim\mathcal{D}|\mathcal{D}\ni i}\left[U(S)\right] - \mathbb{E}_{S\sim\mathcal{D}|\mathcal{D}\not\ni i}\left[U(S)\right]$? The answer is unfortunately negative. Note that in order to write $\phi_{\text{semi}}\left(i; U, N, w\right)$ in this way, we must have

$$\begin{aligned}
\Pr[\mathcal{D} = S | i\in S] &= \frac{1}{n}w(|S|+1) \\
\Pr[\mathcal{D} = S | i\notin S] &= \frac{1}{n}w(|S|)
\end{aligned}$$

for any $i\in N$. Now, we consider a particular S s.t. $i\notin S, j\in S$. Denote $x = \Pr[i\notin\mathcal{D}]$, $y = \Pr[j\notin\mathcal{D}]$, and $k = |S|+1$. By Bayes theorem, we have

$$\begin{aligned}
\Pr[\mathcal{D} = S | i\notin S, j\in S] &= \frac{\Pr[j\in\mathcal{D}|\mathcal{D}=S, i\notin\mathcal{D}]\Pr[\mathcal{D}=S|i\notin\mathcal{D}]}{\Pr[j\in\mathcal{D}]} \\
&= \frac{w(k-1)}{n(1-y)} \\
&= \frac{\Pr[i\notin\mathcal{D}|\mathcal{D}=S, j\in\mathcal{D}]\Pr[\mathcal{D}=S|j\in\mathcal{D}]}{\Pr[i\notin\mathcal{D}]} \\
&= \frac{w(k)}{nx}
\end{aligned}$$

Thus we have $w(k-1)x = w(k)(1-y)$. Similarly, consider a S' of the same size s.t. $i \in S, j \notin S$, we obtain $w(k-1)y = w(k)(1-x)$.

Given

$$w(k-1)x = w(k)(1-y)$$
$$w(k-1)y = w(k)(1-x)$$

If $x = y$, then we have $x = \frac{w(k-1)}{w(k-1)+w(k)}$ which clearly depends on k unless $w(1), w(2), \ldots, w(n)$ is a geometric series ($w(k-1) + w(k)$ cannot be 0 for all k, and x can also not be 0). The only known semivalue-based data valuation method that satisfies this property is the Banzhaf value.

If $x \neq y$, then we have $w(k-1)(x-y) = w(k)(x-y)$, which clearly leads to $w(k-1) = w(k)$ where Banzhaf value is still the only choice.

C.3 Robustness of MSR Estimator

Robustness of MSR Estimator Under Noisy Utility Function. As discussed in Section 3, the utility function U is re-defined as the expected model performance due to the stochasticity of the underlying learning algorithm. Hence, the actual estimator of the Banzhaf value that we build is based upon the noisy variant $\widehat{U}$:

$$\widetilde{\phi}_{\mathrm{MSR}}(i) = \frac{1}{|\mathcal{S}_{\ni i}|} \sum_{S \in \mathcal{S}_{\ni i}} \widehat{U}(S) - \frac{1}{|\mathcal{S}_{\not\ni i}|} \sum_{S \in \mathcal{S}_{\not\ni i}} \widehat{U}(S). \tag{15}$$

Hence, it is interesting to understand the impact of noisy utility function evaluation on the sample complexity of the MSR estimator.

Theorem C.2. *When $\|U - \widehat{U}\|_2 \leq \gamma$, $\widetilde{\phi}_{\mathrm{MSR}}$ is $(\varepsilon + \frac{\gamma\sqrt{n}}{2^{n/2-1}}, \delta)$-approximation in ℓ_2-norm with $O(\frac{n}{\varepsilon^2}\log(\frac{n}{\delta}))$ calls, and $(\varepsilon + \frac{\gamma}{2^{n/2-1}}, \delta)$-approximation in ℓ_∞-norm with $O(\frac{1}{\varepsilon^2}\log(\frac{n}{\delta}))$ calls to $\widehat{U}$.*

The theorem above shows that our MSR algorithm has the same sample complexity in the presence of noise in $\widehat{U}$, with a small extra irreducible error since typically $\gamma \propto \sqrt{2^n}$.

Recall that

$$\widetilde{\phi}_{\mathrm{MSR}}(i) = \frac{1}{|\mathcal{S}_{\ni i}|} \sum_{S \in \mathcal{S}_{\ni i}} \widehat{U}(S) - \frac{1}{|\mathcal{S}_{\not\ni i}|} \sum_{S \in \mathcal{S}_{\not\ni i}} \widehat{U}(S)$$

Theorem C.2. When $\|U - \widehat{U}\|_2 \leq \gamma$, $\widetilde{\phi}_{\mathrm{MSR}}$ is $(\varepsilon + \frac{\gamma\sqrt{n}}{2^{n/2-1}}, \delta)$-approximation in ℓ_2-norm with $O(\frac{n}{\varepsilon^2}\log(\frac{n}{\delta}))$ calls, and $(\varepsilon + \frac{\gamma}{2^{n/2-1}}, \delta)$-approximation in ℓ_∞-norm with $O(\frac{1}{\varepsilon^2}\log(\frac{n}{\delta}))$ calls to $\widehat{U}$.

Proof. Note that for each $i \in N$,

$$\left|\widetilde{\phi}(i) - \phi(i)\right| \leq \left|\widetilde{\phi}(i) - \widehat{\phi}(i)\right| + \left|\widehat{\phi}(i) - \phi(i)\right|$$

From Theorem 4.9, we have

$$\Pr_{\widehat{\phi}}\left[\left|\widehat{\phi}(i) - \phi(i)\right| \geq \Delta\right] \leq 5\exp\left(-\frac{m}{32}\Delta^2\right)$$

Now we bound $\left|\widetilde{\phi}(i) - \widehat{\phi}(i)\right|$.

$$\begin{aligned}
\left|\widetilde{\phi}(i) - \widehat{\phi}(i)\right| &= \left|\frac{1}{|\mathcal{S}_{\ni i}|}\sum_{S \in \mathcal{S}_{\ni i}}\left(U(S) - \widehat{U}(S)\right) - \frac{1}{|\mathcal{S}_{\not\ni i}|}\sum_{S \in \mathcal{S}_{\not\ni i}}\left(U(S) - \widehat{U}(S)\right)\right| \\
&\leq \frac{1}{|\mathcal{S}_{\ni i}|}\sum_{S \in \mathcal{S}_{\ni i}}\left|U(S) - \widehat{U}(S)\right| + \frac{1}{|\mathcal{S}_{\not\ni i}|}\sum_{S \in \mathcal{S}_{\not\ni i}}\left|U(S) - \widehat{U}(S)\right|
\end{aligned}$$

Given $\left\|U - \widehat{U}\right\| \leq \gamma$, we bound $\left|\widetilde{\phi}(i) - \widehat{\phi}(i)\right|$ as follows:

$$\begin{aligned}
&\Pr\left[\left|\widetilde{\phi}(i) - \widehat{\phi}(i)\right| \geq \varepsilon\right] \\
&\leq \Pr\left[\frac{1}{|\mathcal{S}_{\ni i}|}\sum_{S \in \mathcal{S}_{\ni i}}\left|U(S) - \widehat{U}(S)\right| + \frac{1}{|\mathcal{S}_{\not\ni i}|}\sum_{S \in \mathcal{S}_{\not\ni i}}\left|U(S) - \widehat{U}(S)\right| \geq \varepsilon\right] \\
&\leq \Pr\left[\frac{1}{|\mathcal{S}_{\ni i}|}\sum_{S \in \mathcal{S}_{\ni i}}\left|U(S) - \widehat{U}(S)\right| + \frac{1}{|\mathcal{S}_{\not\ni i}|}\sum_{S \in \mathcal{S}_{\not\ni i}}\left|U(S) - \widehat{U}(S)\right| \geq \varepsilon \middle| \left||\mathcal{S}_{\ni i}| - \frac{m}{2}\right| \leq \Delta\right] + 2\exp\left(-\frac{2\Delta^2}{m}\right) \\
&\leq \frac{\Pr\left[\frac{2}{m}\sum_{S \subseteq \mathcal{S}}\left|U(S) - \widehat{U}(S)\right| \geq \varepsilon - \frac{4\Delta}{m}\right]}{1 - 2\exp\left(-\frac{2\Delta^2}{m}\right)} + 2\exp\left(-\frac{2\Delta^2}{m}\right)
\end{aligned} \tag{16}$$

By $\left\|U-\widehat{U}\right\| \leq \gamma$, we have

$$\mathbb{E}[|U(S)-\widehat{U}(S)|] = \frac{1}{2^n}\sum_{S\subseteq N}\left|U(S)-\widehat{U}(S)\right| = \frac{1}{2^n}\left\|U-\widehat{U}\right\|_1 \leq \frac{\sqrt{2^n}}{2^n}\left\|U-\widehat{U}\right\| = \frac{\gamma}{2^{n/2}}$$

Set $\varepsilon' = \varepsilon - \frac{\gamma}{2^{n/2-1}}$. Thus

$$\begin{aligned}
(16) &= \frac{\Pr\left[\frac{2}{m}\sum_{S\subseteq\mathcal{S}}\left|U(S)-\widehat{U}(S)\right| - \frac{\gamma}{2^{n/2-1}} \geq \varepsilon' - \frac{4\Delta}{m}\right]}{1-2\exp\left(-\frac{2\Delta^2}{m}\right)} + 2\exp\left(-\frac{2\Delta^2}{m}\right)\\
&\leq \frac{\Pr\left[\frac{2}{m}\sum_{S\subseteq\mathcal{S}}\left|U(S)-\widehat{U}(S)\right| - 2\mathbb{E}[|U(S)-\widehat{U}(S)|] \geq \varepsilon' - \frac{4\Delta}{m}\right]}{1-2\exp\left(-\frac{2\Delta^2}{m}\right)} + 2\exp\left(-\frac{2\Delta^2}{m}\right)\\
&\leq \frac{\Pr\left[\left|\frac{2}{m}\sum_{S\subseteq\mathcal{S}}\left|U(S)-\widehat{U}(S)\right| - \frac{\gamma}{2^{n/2-1}}\right| \geq \varepsilon' - \frac{4\Delta}{m}\right]}{1-2\exp\left(-\frac{2\Delta^2}{m}\right)} + 2\exp\left(-\frac{2\Delta^2}{m}\right)\\
&\leq \frac{2\exp\left(-\frac{1}{8}m\left(\varepsilon'-\frac{4\Delta}{m}\right)^2\right)}{1-2\exp\left(-\frac{2\Delta^2}{m}\right)} + 2\exp\left(-\frac{2\Delta^2}{m}\right)\\
&\leq 5\exp\left(-\frac{m}{32}(\varepsilon')^2\right)
\end{aligned}$$

Thus, we have

$$\Pr_{\widehat{\phi}}\left[\left|\widehat{\phi}(i)-\widetilde{\phi}(i)\right| \geq \varepsilon + \frac{\gamma}{2^{n/2-1}}\right] \leq 5\exp\left(-\frac{m}{32}\varepsilon^2\right)$$

Therefore,

$$\begin{aligned}
\Pr\left[\left|\widetilde{\phi}(i)-\phi(i)\right| \geq \varepsilon + \frac{\gamma}{2^{n/2-1}}\right] &\leq \Pr\left[\left|\widetilde{\phi}(i)-\widehat{\phi}(i)\right| + \left|\widehat{\phi}(i)-\phi(i)\right| \geq \varepsilon + \frac{\gamma}{2^{n/2-1}}\right]\\
&= 1 - \Pr\left[\left|\widetilde{\phi}(i)-\widehat{\phi}(i)\right| + \left|\widehat{\phi}(i)-\phi(i)\right| \leq \varepsilon + \frac{\gamma}{2^{n/2-1}}\right]\\
&\leq 1 - \Pr\left[\left|\widetilde{\phi}(i)-\widehat{\phi}(i)\right| \leq \frac{\varepsilon}{2} + \frac{\gamma}{2^{n/2-1}} \wedge \left|\widehat{\phi}(i)-\phi(i)\right| \leq \frac{\varepsilon}{2}\right]\\
&\leq \Pr\left[\left|\widetilde{\phi}(i)-\widehat{\phi}(i)\right| \geq \frac{\varepsilon}{2} + \frac{\gamma}{2^{n/2-1}}\right] + \Pr\left[\left|\widehat{\phi}(i)-\phi(i)\right| \geq \frac{\varepsilon}{2}\right]\\
&= 10\exp\left(-\frac{m}{128}\varepsilon^2\right)
\end{aligned}$$

By union bound, we have

$$\Pr\left[\left\|\widetilde{\phi}_{\mathrm{MSR}}-\phi\right\|_2 \geq \varepsilon + \frac{\gamma\sqrt{n}}{2^{n/2-1}}\right] \leq 10n\exp\left(-\frac{m\varepsilon^2}{128n}\right)$$

and

$$\Pr\left[\left\|\widetilde{\phi}_{\mathrm{MSR}}-\phi\right\|_\infty \geq \varepsilon + \frac{\gamma}{2^{n/2-1}}\right] \leq 10n\exp\left(-\frac{m\varepsilon^2}{128}\right)$$

By setting $\delta \leq 10n\exp\left(-\frac{m\varepsilon^2}{128n}\right)$, we obtain the sample complexity $O\left(\frac{n}{\varepsilon^2}\log(\frac{n}{\delta})\right)$ for ℓ_2-norm, and by setting $\delta \leq 10n\exp\left(-\frac{m\varepsilon^2}{128}\right)$, we obtain the sample complexity $O\left(\frac{1}{\varepsilon^2}\log(\frac{n}{\delta})\right)$ for ℓ_∞-norm. □

C.4 Stability of Banzhaf value in ℓ_2-norm

As mentioned previously, we can alternatively view a semivalue as a function $\phi : \mathbb{R}^{2^n} \to \mathbb{R}^n$ which takes a utility function $U \in \mathbb{R}^{2^n}$ as input, and output the values of data points $\phi(U) \in \mathbb{R}^n$. By taking this functional view, a natural robustness measure for semivalue $\phi(\cdot; w)$ is its Lipschitz constant L, which is defined as the smallest constant such that

$$\left\|\phi(U; w) - \phi(\widehat{U}; w)\right\| \leq L \left\|U - \widehat{U}\right\|$$

for all possible pairs of U and $\widehat{U}$.

Theorem C.3. *Among all semivalues, Banzhaf value ($w(k) = \frac{n}{2^{n-1}}$) achieves the smallest Lipschitz constant $L = \frac{1}{2^{n/2-1}}$. In other words, for the Banzhaf value we have*

$$\left\|\phi_{\text{banz}}(U) - \phi_{\text{banz}}(\widehat{U}; w)\right\| \leq \frac{1}{2^{n/2-1}} \left\|U - \widehat{U}\right\|$$

for all possible pairs of U and $\widehat{U}$, and $L = \frac{1}{2^{n/2-1}}$ is the smallest constant among all semivalues.

Proof. Recall that a semivalue has the following representation

$$\phi_{\text{semi}}\left(i; U, w\right) := \frac{1}{n} \sum_{k=1}^{n} w(k) \sum_{S \subseteq N \setminus \{i\}, |S| = k-1} \left[U(S \cup i) - U(S)\right]$$

An interesting observation about semivalue is that the transformation $\phi : \mathbb{R}^{2^n} \to \mathbb{R}^n$ is always a linear transformation. Thus, for every semivalue, we can define *Semivalue matrix* $S_n \in \mathbb{R}^{n \times 2^n}$ where $\phi(U) = S_n U$. We denote the ith row of S_n as $(S_n)_i$, and the entry in the ith row corresponding to subset S as $(S_n)_{i,S}$. It is not hard to see that

$$(S_n)_{i,S} = \frac{1}{n} w(|S|) \text{ if } i \in S$$
$$(S_n)_{i,S} = -\frac{1}{n} w(|S| + 1) \text{ if } i \notin S$$

The Lipschitz constant of ϕ is thus equal to the operator norm of matrix S_n, which is the square root of the largest eigenvalue of matrix $S_n S_n^T$. Now we compute the eigenvalue of matrix $S_n S_n^T$.

For matrix $S_n S_n^T$, its diagonal entry is

$$\begin{aligned} d_1 &= \sum_{S \in 2^n, i \in S} \frac{1}{n^2} w^2(|S|) + \sum_{S \in 2^n, i \notin S} \frac{1}{n^2} w^2(|S| + 1) \\ &= \frac{1}{n^2} \left[\sum_{k=1}^{n} \binom{n-1}{k-1} w^2(k) + \sum_{k=1}^{n} \binom{n-1}{k-1} w^2(k) \right] \\ &= \frac{2}{n^2} \sum_{k=1}^{n} \binom{n-1}{k-1} w^2(k) \end{aligned}$$

and its non-diagonal entry is

$$\begin{aligned} d_2 &= \sum_{k=2}^{n} \binom{n-2}{k-2} \left(\frac{1}{n} w(k)\right)^2 + 2 \sum_{k=1}^{n-1} \binom{n-2}{k-1} \left(-\frac{1}{n^2} w(k) w(k+1)\right) + \sum_{k=0}^{n-2} \binom{n-2}{k} \left(\frac{1}{n} w(k+1)\right)^2 \\ &= \frac{1}{n^2} \sum_{k=0}^{n-2} \binom{n-2}{k} \left[w^2(k+2) - 2w(k+1)w(k+2) + w^2(k+1)\right] \\ &= \frac{1}{n^2} \sum_{k=0}^{n-2} \binom{n-2}{k} \left(w(k+2) - w(k+1)\right)^2 \end{aligned}$$

Therefore we can write $S_n S_n^T = (d_1 - d_2)\mathbb{1}_n + d_2 \mathbf{1}_n$ where $\mathbb{1}_n \in \mathbb{R}^{n \times n}$ is the identity matrix and $\mathbf{1}_n \in \mathbb{R}^n$ is all-one matrix. The two eigenvalues are $d_1 + (n-1)d_2$ and $d_1 - d_2$. Since $d_2 \geq 0$, the top eigenvalue is $d_1 + (n-1)d_2$. Therefore, our goal is to find weight function w such that

$$\begin{aligned} &\min_w d_1 + (n-1)d_2 \\ &\text{subject to } \sum_{k=1}^{n} \binom{n-1}{k-1} w(k) = n \end{aligned}$$

that is, we want to solve

$$\begin{aligned} &\min_w \frac{2}{n^2} \sum_{k=1}^{n} \binom{n-1}{k-1} w(k)^2 + \frac{n-1}{n^2} \sum_{k=1}^{n-1} \binom{n-2}{k-1} \left(w(k) - w(k+1)\right)^2 \\ &\text{subject to } \sum_{k=1}^{n} \binom{n-1}{k-1} w(k) = n \end{aligned}$$

Note that by Cauchy-Schwarz inequality,

$$\begin{aligned} n^2 &= \left(\sum_{k=1}^{n} \binom{n-1}{k-1} w(k) \right)^2 \\ &= \left(\sum_{k=1}^{n} \sqrt{\binom{n-1}{k-1}} \sqrt{\binom{n-1}{k-1}} w(k) \right)^2 \\ &\leq \left(\sum_{k=1}^{n} \binom{n-1}{k-1} \right) \left(\sum_{k=1}^{n} \binom{n-1}{k-1} w(k)^2 \right) \\ &= 2^{n-1} \sum_{k=1}^{n} \binom{n-1}{k-1} w(k)^2 \end{aligned}$$

Thus the first term in the objective function is lower bounded by $\frac{2}{2^{n-1}}$, which is achieved when $w(1) = \ldots = w(n) = \frac{n}{2^{n-1}}$, and this is also where the minimum of the second term achieved, i.e., just 0. Thus, the minimum possible $\min_w = d_1 + (n-1)d_2 = \frac{1}{2^{n-2}}$, and thus the operator norm of $S_n = \sqrt{\frac{1}{2^{n-2}}} = \frac{1}{2^{n/2-1}}$. □

D EXPERIMENT SETTINGS & ADDITIONAL EXPERIMENTAL RESULTS

We provide a summary of the content in this section for the convenience of the readers.

- Appendix D.1: Experiment Settings for Figure 1 and 2 in the main text.
- Appendix D.2: Experiment Settings for Sample Efficiency Experiment in Section 5.1.
- Appendix D.3: Experiment Settings for Applications in Section 5.3.
- Appendix D.4: Experiment Settings and Additional Results for Rank Stability Experiment in Section 5.2.
- Appendix D.5: Additional Results for Rank Stability Experiment on Tiny Datasets.
- Appendix D.6: Additional Results for Rank Stability on Gradient Descent with Randomized Smoothing.

D.1 Experiment Settings for Figure 1 and 2 in the Main Text.

We estimate the LOO error, Shapley, and Banzhaf value on a size-2000 CIFAR10 dataset where we randomly flip the label of 10% data points. We use the state-of-the-art estimator for the Shapley and Banzhaf value for Figure 1 (b) and (c), i.e., Permutation sampling for the Shapley value and our MSR estimator for Banzhaf value. For both the Shapley and Banzhaf value, we set the number of samples as 50,000 where the sampling distributions depend on the corresponding estimators. We compute the LOO with its exact formula but with noisy utility scores, and we align the number of (potentially repeated) samples also as 50,000. For each sample S, we train 5 models on it with different random seeds, and obtain 5 noisy versions of $U(S)$. We then compute 5 different LOO/Shapley/Banzhaf values for 20 randomly selected CIFAR10 images (including 5 mislabeled images), and draw the corresponding box-plot in Figure 1. The learning architecture we use is a standard CNN adapted from PyTorch tutorial[8], with batch size 32, learning rate 10^{-3} and Adam optimizer for training.

In Figure 2, we compare the stability of different data valuation techniques in maintaining a consistent set of top-influence data points. Specifically, we follow the same protocol as the setting for Figure 1 and compute 5 different versions of data value scores for each data point. We then count the percentage of data points that are consistently ranked in the top or bottom-$k\%$ across all the runs.

Remark D.1. The results in Figure 1 and 2 in the main text also depend on the robustness of data valuation methods' corresponding estimator. We use the best-known estimators for each data valuation method in the experiment to reduce the impact of the estimation procedure as much as possible.

D.2 Experiment Settings for Sample Efficiency Experiment in Section 5.1

For Figure 4 (a), we use a synthetic dataset with only 10 data points. To generate the synthetic dataset, we sample 10 data points from a bivariate Gaussian distribution where the means are 0.1 and -0.1 on each dimension, and the covariance matrix is the identity matrix. The labels are assigned to be the sign of the sum of the two features. The utility of a subset is the test accuracy of the model trained on the subset. A logistic regression classifier trained on the 10 data points achieves around 80% test accuracy. We show the Banzhaf value estimate for *one* data point in Figure 4 (a).

For Figure 4 (b), we use a size-500 MNIST dataset. The Relative Spearman Index is computed by the Spearman Index of the ranking of the value estimates between the current iteration, and the ranking of the value estimates when given additional 1000 samples. The learning architecture we use is LeNet (LeCun et al., 1989), with batch size 32, (initial) learning rate 10^{-3} and Adam optimizer for training.

D.3 Experiment Settings for Applications in Section 5.3

D.3.1 Datasets & Models

A comprehensive list of datasets and sources is summarized in Table 3. Similar to the existing data valuation literature (Ghorbani and Zou, 2019; Kwon and Zou, 2021; Jia et al., 2019b; Wang et al., 2021), we preprocess datasets for the ease of training. For Fraud, Creditcard, Vehicle, and all datasets from OpenML, we subsample the dataset to balance positive and negative labels. For these datasets, if they have multi-class, we binarize the label by considering $\mathbb{1}[y = 1]$. For the image

[8] https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

Dataset	Source
MNIST	LeCun (1998)
FMNIST	Xiao et al. (2017)
CIFAR10	(Krizhevsky et al., 2009)
Click	https://www.openml.org/d/1218
Fraud	Dal Pozzolo et al. (2015)
Creditcard	Yeh and Lien (2009)
Vehicle	Duarte and Hu (2004)
Apsfail	https://www.openml.org/d/41138
Phoneme	https://www.openml.org/d/1489
Wind	https://www.openml.org/d/847
Pol	https://www.openml.org/d/722
CPU	https://www.openml.org/d/761
2DPlanes	https://www.openml.org/d/727

Table 3: A summary of datasets used in Section 5.3's experiments.

dataset CIFAR10, we follow the common procedure in prior works (Ghorbani and Zou, 2019; Jia et al., 2019b; Kwon and Zou, 2021): we extract the penultimate layer outputs from the pre-trained ResNet18 (He et al., 2016). The pre-training is done with the ImageNet dataset (Deng et al., 2009) and the weight is publicly available from PyTorch. We choose features from the class of Dog and Cat. The extracted outputs have dimension 512. For the image dataset MNIST and FMNIST, we directly train on the original data format, which is a more challenging setting compared with the previous literature.

For MNIST and FMNIST, we use LeNet (LeCun et al., 1989), with batch size 128, (initial) learning rate 10^{-3} and Adam optimizer for training. For CIFAR10 dataset, we use a two-layer MLP where there are 256 neurons in the hidden layer, with activation function ReLU, with batch size 128, (initial) learning rate 10^{-3} and Adam optimizer for training. For the rest of the datasets, we use a two-layer MLP where there are 100 neurons in the hidden layer, with activation function ReLU, (initial) learning rate 10^{-2}, and Adam optimizer for training. We use batch size 128 for Creditcard, Apsfail, Click, and CPU dataset, and batch size 32 for the rest of datasets.

D.3.2 Experiment Settings

For MNIST, FMNIST, and CIFAR10, we consider the number of data points being valued as 2000. For Click dataset, we consider the number of data points to be valued as 1000. For Phoneme dataset, we consider the number of data points to be valued as 500. For the rest of the datasets, we consider the number of data points to be valued as 200. For each data value we show in Table 1 and 2, we use the corresponding state-of-the-art estimator to estimate them (for Data Shapley, we use Permutation Sampling; for Data Banzhaf, we use our MSR estimator; for Beta Shapley, we use the Monte Carlo estimator by Kwon and Zou (2021)). We stress that the Monte Carlo estimator by Kwon and Zou (2021) is not numerically stable when the training set size > 500, so for datasets with > 500 data points (MNIST, FMNIST, CIFAR10, and Click), we omit the results for Beta Shapley. We set the number of samples to estimate Data Banzhaf, Data Shapley, and Beta Shapley as 100,000.

All of our experiments are performed on Tesla P100-PCIE-16GB GPU.

Learning with Weighted Samples. For each estimated data value, we normalize it to $[0, 1]$ by $\frac{\texttt{value}-\texttt{min}}{\texttt{max}-\texttt{min}}$. Let $\phi(i)$ be the *normalized* value for data point i. We compare the test accuracy of a weighted risk minimizer f_ϕ defined as

$$f_\phi := \operatorname*{argmin}_f \sum_{i \in N} \phi(i) \text{loss}_f(i) \tag{17}$$

where $\text{loss}_f(i)$ denote the loss of f on data point $i \in N$.

Noisy Label Detection. We flip 10% of the labels by picking an alternative label from the rest of the classes uniformly at random.

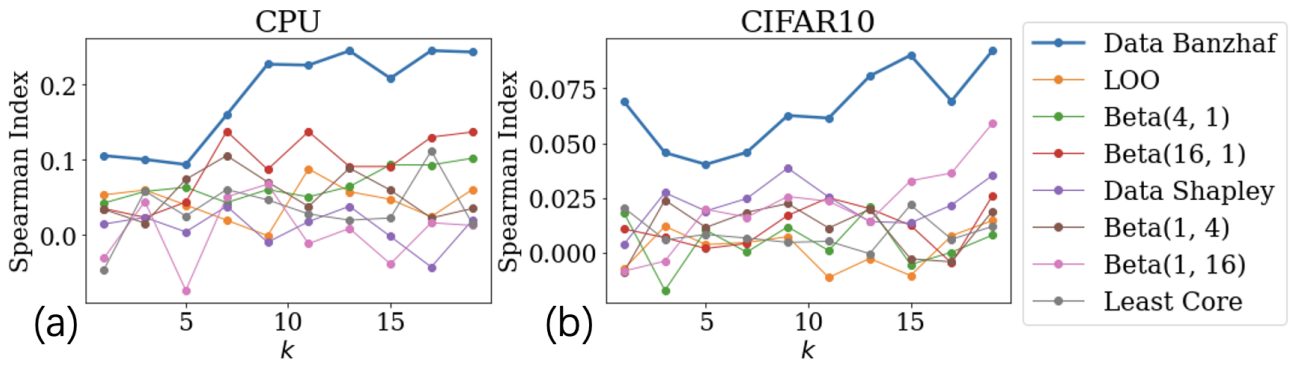

Figure 6: The stability of data value ranking measured by Spearman index between the ranking of "ground-truth" data value and the ranking of data value estimated from noisy utility scores, where (a) is on CPU dataset and (b) is on CIFAR10 dataset (same as Figure 5 except for the additional curve for the Least core).

D.4 Experiment Settings and Additional Results for Rank Stability Experiment in Section 5.2

In this section, we describe the detailed settings and additional results on comparing the rank stability of different data values on natural datasets. We also evaluate an additional baseline, the *least cores*, another existing data value notion which is *not* a semivalue but also originates from cooperative game theory (see the description in Appendix A). The estimation algorithm for the least core is the Monte Carlo algorithm from Yan and Procaccia (2020).

Settings. We experiment on 'CPU' (200 data points) and 'CIFAR10' (500 data points) datasets from Table 3. The data preprocessing procedure, model training hyperparameters, and the estimation algorithm for semivalues are the same as what have described in Appendix D.3.1 and D.3.2. The perturbations of the model performance scores are caused by the randomness in neural network initialization and mini-batch selection in SGD.

The tricky part of the experiment design is that we need to find a way to adjust the scale of the perturbation caused by a natural stochastic learning algorithm. Our preliminary experiments show that the variance of performance scores does not have a clear dependency on the training hyperparameters such as mini-batch sizes. To solve this challenge, we design the following procedure to control the magnitude of the perturbation with a single parameter k:

1. Sample m data subsets $S_1, \ldots, S_m$ (the subset sampling distributions depend on specific semivalue estimators).
2. For each subset S_i, we execute $\widehat{U}(S_i)$ for k times and obtain k independent performance score samples $u_1, \ldots, u_k \sim \widehat{U}(S_i)$. We compute $\widetilde{U}_k(S_i) = \frac{1}{k}\sum_{j=1}^{k} u_j$.
3. Estimate the semivalue with the corresponding estimators based on samples $\widetilde{U}_k(S_1), \ldots, \widetilde{U}_k(S_m)$.

In other words, k is the number of runs that we execute a stochastic learning algorithm in order to estimate the expected utility on a given subset. With a larger k, the noise in the estimated utility will become smaller. Since it is infeasible to compute the ground-truth data values for this experiment, we approximate the ground-truth by setting $k = 50$ (called *reference data value* in the caption of figures). As k increases, the difference between $\widehat{U}_k(S_i)$ and the approximated ground-truth scores will be smaller with high probability. We set the budget of samples used to estimate the semivalues as $m = 2000$ (same for the ground-truth) for all semivalues for a fair comparison. It is worth noting that in this case, the rank stability is not just related to the property of the data value notion, but also the corresponding estimator.

Results. We plot the Spearman index between the approximated ground-truth data value ranking and the estimated data value ranking with different ks in Figure 6, where (b) is the same figure we show in the main text (with the additional baseline of the Least core). As we can see, Data Banzhaf once again outperforms all other data value notions on both datasets. It achieves better rank stability than others by a large margin for a wide range of ks.

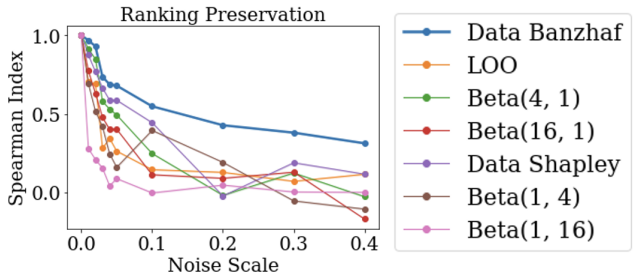

Figure 7: Impact of the noise in utility scores on the stability of data value ranking measured by Spearman index between the ranking of exact data value and the ranking of data value estimated from noisy utility scores.

D.5 Additional Results for Ranking Stability Experiment on Tiny Datasets

The ranking stability results in Section 5.2 and Appendix D.4 do not compute the exact data value but use the corresponding estimation algorithms. In this section, we present an additional result of ranking stability on tiny datasets when we are able to compute the exact data value.

Similar to the evaluation protocol for sample efficiency comparison, we experiment on a synthetic dataset with a scale (10 data points) that we can compute the exact ranking for different data value notions. We use the same synthetic dataset as in the sample efficiency experiment in Section 5.1. The performance score of a subset is the test accuracy of the Logistic regression model trained on the subset. In Figure 7, we plot the Spearman index between the ranking of exact data values and the ranking of data values computed from noisy utility scores. For each noise scale σ on the x-axis of Figure 7 (a), we add random Gaussian noise $\mathcal{N}(0, \sigma\mathbf{I})$ to perturb the performance score. That is, $\widehat{U} = U + \mathcal{N}(0, \sigma\mathbf{I})$. We then compute the Spearman index between the ranking of exact data values (derived from U) and the ranking of data values derived from noisy utility scores $\widehat{U}$. We repeat this procedure 20 times and take the average Spearman index for each point in the figure.

The main considerations behind the design choices of synthetic dataset and Gaussian noise addition are the following:

- In order to rule out the influence of estimation error, we would like to compute the *exact* ranking of data points in terms of different data value notions, which means that we can only use a toy example with ≤ 15 data points. In this case, it does not make sense to use SGD for training.
- According to our preliminary experiment results, the variance of performance scores does not have a clear dependency on the SGD's training hyperparameters such as mini-batch sizes. The relationship between performance variance and training hyperparameters is an interesting direction for future work.

D.6 Additional Results for Rank Stability on Gradient Descent with Randomized Smoothing

In our experiment, we mainly use SGD and its variants as the test case since SGD is arguably the most frequently used stochastic learning algorithm nowadays. However, the robustness guarantee derived in our theory (Section 4) is agnostic to the structure of perturbation, which means that it applies to perturbations caused by arbitrary kinds of learning algorithms. Therefore, we expect to get similar rank stability results when experimenting with other kinds of stochastic learning algorithms. For completeness, we perform an additional ranking stability experiment with another useful stochastic learning algorithm, *gradient descent with randomized smoothing* (Duchi et al., 2012).

We use $\text{loss}(\theta, N) = \sum_{i \in N} \text{loss}(\theta, i)$ to denote the loss of model with parameter θ on the dataset N. For regular gradient descent, at iteration t, the model is updated as

$$\theta_{t+1} = \theta_t - \eta \nabla \text{loss}(\theta_t, N) \tag{18}$$

where ∇ denotes the derivative with respect to θ. In contrast to the regular gradient descent, the randomized smoothing technique convolves Gaussian noise with the original learning loss function and the model is updated instead by "smoothed gradient":

$$\theta_{t+1} = \theta_t - \eta \frac{1}{\ell} \sum_{j=1}^{\ell} \nabla \text{loss}(\theta_t + \alpha \mathcal{N}(0, \mathbf{I}), N) \tag{19}$$

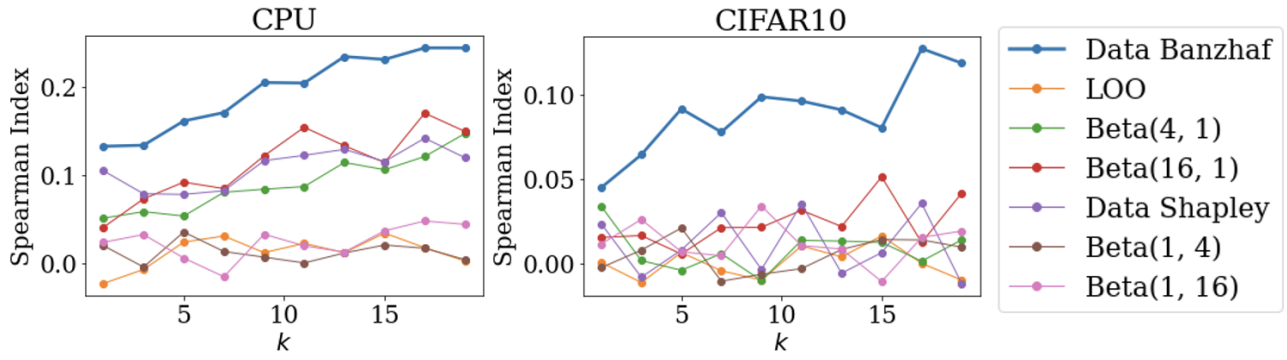

Figure 8: The stability of data value ranking measured by Spearman index between the ranking of reference data value and the ranking of data value estimated from noisy utility scores, where (a) is on CPU dataset and (b) is on CIFAR10 dataset.

We compare the rank stability of different semivalues on the CPU dataset and the CIFAR10 dataset, with exactly the same experiment setting as in Appendix D.4, except for replacing SGD-based training with the gradient descent with randomized smoothing technique. We set $\ell = 1$ to introduce larger randomness, and the smoothing radius α to be equal to the learning rate. The results are shown in Figure 8. As we can see, Data Banzhaf once again outperforms all other data value notions when the sources of performance score perturbation are changed from SGD to gradient descent with randomized smoothing.