Kernel Banzhaf: A Fast and Robust Estimator for Banzhaf Values

Anonymous authors
Paper under double-blind review

Abstract

Banzhaf values provide a popular, interpretable alternative to the widely-used Shapley values for applications in explainable AI. Like Shapley values, computing Banzhaf values exactly requires time exponential in the number of inputs, necessitating the use of efficient estimators. In this work, we introduce Kernel Banzhaf, a regression-based estimator for Banzhaf values. Our approach leverages an existing regression formulation, whose exact solution corresponds to the exact Banzhaf values. Inspired by the success of Kernel SHAP for Shapley values, Kernel Banzhaf obtains an efficient approximation to the Banzhaf values by solving a subsampled instance of this regression problem. Through empirical evaluations across eight datasets, we find that Kernel Banzhaf significantly outperforms prior methods in terms of accuracy and robustness to noise. We complement our experimental evaluation with strong theoretical guarantees on Kernel Banzhaf's performance.

1 Introduction

The increasing complexity of AI models has intensified challenges associated with model interpretability. Modern models, such as deep neural networks and ensemble methods, often make opaque decisions whose justification is not readily apparent to the user. This opacity makes it difficult to trust or rely on model predictions, especially in decision-making scenarios like healthcare, finance, and legal applications, which require rigorous justifications. Thus, there is a pressing need for reliable explainability tools to bridge the gap between complex model behaviors and human understanding.

Among the various methods employed within explainable AI, game-theoretic approaches have gained prominence for quantifying the contribution of data features, both to the overall performance of a machine learning model, and to individual predictions made by the model. The most well-known game-theoretic approach is based on *Shapley values*, which provide a principled way to attribute the contribution of n individual players to the outcome of a cooperative game, which is defined by a set function v that maps every subset of $\{1, \ldots, n\}$ to a real value (Shapley, 1953).

In the context of feature attribution, each "player" is a feature and v maps a subset of features to, e.g., the overall test loss when the chosen features are used, or the prediction made for a specific individual given access to the chosen features. The Shapley value quantifies the average marginal contribution of a feature on the set function, computed as the weighted average over all possible combinations of features included in the model (Lundberg & Lee, 2017). More formally, the Shapley value ϕ_i of a player $i \in \{1, \ldots, n\}$ is

$$\phi_i^{\text{shap}} = \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} [v(S \cup \{i\}) - v(S)]. \tag{1}$$

While primarily associated with feature attribution (Lundberg & Lee, 2017; Karczmarz et al., 2022), game-theoretic methods like Shapley values are also used in machine learning tasks like feature selection (Covert et al., 2020) and data valuation (Ghorbani & Zou, 2019; Wang & Jia, 2023).

An alternative to Shapley values are Banzhaf values, which also compute each individual's contribution to an overall outcome (Banzhaf, 1965; Lehrer, 1988). While Shapley values are more widely used, Banzhaf values

are often considered more intuitive for AI applications because they treat each subset of players as equally important, instead of weighting by a function of the subset size, as in Equation (1) for Shapley values. In particular, the Banzhaf value is

$$\phi_i^{\text{banz}} = \frac{1}{2^{n-1}} \sum_{S \subseteq [n] \setminus \{i\}} [v(S \cup \{i\}) - v(S)]. \tag{2}$$

It has also been observed that Banzhaf values tend to be more robust than Shapley values in the context of explainable AI (Karczmarz et al., 2022; Wang & Jia, 2023). These benefits have led to the use of Banzhaf values for feature attribution (Datta et al., 2015; Kulynych & Troncoso, 2017; Sliwinski et al., 2018; Patel et al., 2021) and, more recently, for data valuation Wang & Jia (2023); Li & Yu (2024c).

For general set functions, the exact computation of both Shapley and Banzhaf values requires at least 2^n time, where n is the number of features in our problem. In particular, we must evaluate $v(S)$ for every possible subset $S \subseteq \{1, \ldots, n\}$. Accordingly, the problem of *approximating* Shapley values at lower cost has been widely studied, especially for model explanation. A leading estimator for Shapley values is KernelSHAP (Lundberg & Lee, 2017), a model-agnostic technique that leverages a connection to linear regression, approximating Shapley values by solving a sampled weighted least squares problem (Charnes et al., 1988; Lundberg & Lee, 2017). KernelSHAP has been further improved with paired sampling (Covert & Lee, 2020) and leverage score sampling (Musco & Witter, 2025).

In contrast to Shapley values, fewer algorithms have been proposed to approximate Banzhaf values for arbitrary set functions. For tree-structured set functions, such as those corresponding to decision tree based models, exact Banzhaf values can be efficiently computed (Karczmarz et al., 2022). For general set functions, like those corresponding to neural networks models, Monte Carlo sampling (i.e., estimating the sum in Equation (2) via a subsample) can be used to estimate each Banzhaf value separately (Merrill III, 1982; Bachrach et al., 2010). The Maximum Sample Reuse (MSR) algorithm is a variant of Monte Carlo sampling that reuses samples for the estimates of different Banzhaf values (Wang & Jia, 2023). However, even with reuse, the naive Monte Carlo estimator typically requires a large number of samples (which equates to a large number of evaluations of v) to obtain meaningfully accurate estimates for the Banzhaf values.

Our Contributions. In this work, we show that, like Shapley values, Banzhaf values can be approximated far more accurately through *approximate regression*. To do so, we leverage an existing regression formulation (concretely, a linear least squares regression problem involving n variables) whose solution exactly corresponds to the Banzhaf values (Hammer & Holzman, 1992). We show that this problem can be efficiently solved by subsampling rows of the regression problem, each of which corresponds to a subset of players and requires just a single evaluation of the set function v.

The ultimate payoff is a novel estimator, *Kernel Banzhaf*, that far outperforms existing Monte Carlo estimation methods. In particular, we conduct an extensive experimental evaluation of Kernel Banzhaf in Section 4, assessing performance in terms of sample efficiency and robustness to noise in the evaluations of v. Furthermore, we evaluate Kernel Banzhaf in a downstream feature ranking task. Our experiments span diverse model classes, including tree-based models and neural networks. Our findings suggest that Kernel Banzhaf is significantly more accurate and robust than prior Monte Carlo estimators (see e.g., Figure 1). Concretely, for a fixed number of evaluations of v, our method typically improves on the accuracy of Banzhaf values estimation by an order of magnitude or more.

We complement our experimental evaluation of Kernel Banzhaf with strong theoretical guarantees. We prove that, with just $O(n \log n)$ evaluations of v, Kernel Banzhaf returns a solution with bounded squared ℓ_2-norm difference to the exact Banzhaf values (see Corollary 3.3). Obtaining similar theoretical guarantees for Monte Carlo sampling and improvements like Maximum Sample Reuse requires making a strong assumption on the maximum magnitude of $v(S)$. See Section 3.3 for further comparison of theoretical results.

Our main contributions can be summarized as follows:

1. We propose Kernel Banzhaf, a regression-based approximation algorithm for estimating Banzhaf values for general set functions.

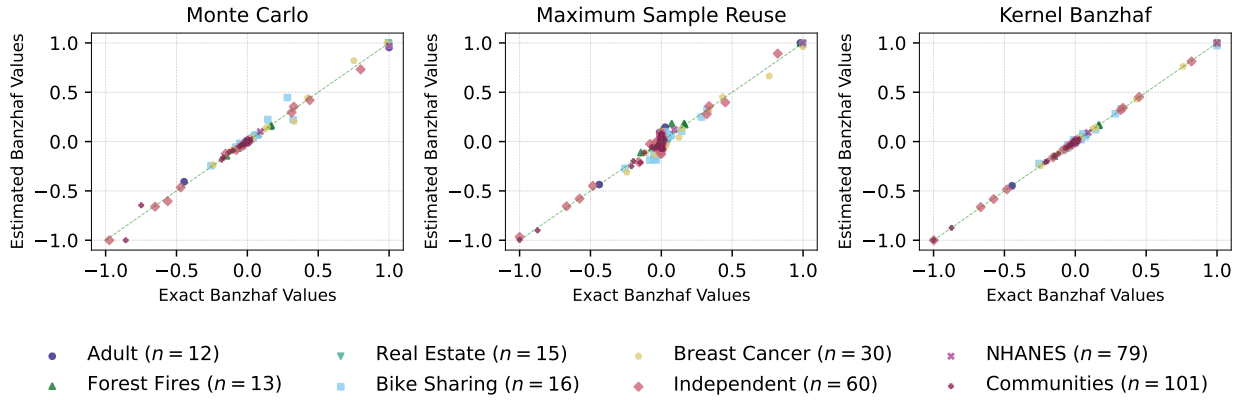

Figure 1: Comparison of exact and estimated Banzhaf feature attribution values across eight datasets, where every estimator uses $m = 20n$ evaluations of the set function v. The evaluations dominate the computational cost of each algorithm. Each plot, labeled with its estimator, shows normalized estimated versus exact Banzhaf values across all features for a randomly selected data point from each dataset. Points closer to the diagonal line indicate more accurate estimates; the plots suggest that Kernel Banzhaf is more accurate than the Monte Carlo and Maximum Sample Reuse estimators.

2. We show that Kernel Banzhaf provably returns accurate solutions. Our theoretical guarantees depend on the magnitude of the true Banzhaf values rather than the magnitude of the set function, as for prior estimators.

3. We extensively evaluate Banzhaf estimator performance across eight popular datasets, in various settings, and under several measures of performance. Our findings demonstrate that Kernel Banzhaf significantly outperforms prior work.

Subsequent Work. Since this preprint was originally posted on arXiv, there has been additional work on efficiently approximating Shapley values and Banzhaf values.One successful approach has been Regression MSR (Witter et al., 2025), which builds on regression-based methods, like Kernel Banzaf, by using them as variance reduction methods for standard Monte Carlo estimators, instead of as direct value estimation algorithms. Indeed, the linear variant of Regression MSR explicitly utilizes Kernel Banzhaf as its surrogate function. Empirically evaluating such recent extensions is outside the scope of this work. Instead, we focus on theoretically and empirically evaluating of the base Kernel Banzhaf estimator, which is a necessary prerequisite to understanding its behavior within hybrid frameworks like Regression MSR.

2 Background

Let n be a positive integer and define $[n] := \{1, 2, \ldots, n\}$. We let $2^{[n]}$ denote the set of all 2^n possible subsets of $[n]$ (including the emptyset). For set S, define the indicator vector $\mathbb{1}[S] \in \{0,1\}^n$ so that $\mathbb{1}[S]_i = 1$ if $i \in S$ and 0 otherwise. Let $v : 2^{[n]} \to \mathbb{R}$ be a set function. Shapley and Banzhaf values are as defined in Equations (1) and (2) in the prior section. For notational simplicity, we will use $\phi_i = \phi_i^{\text{banz}}$ in the rest of the paper. Let $\boldsymbol{\phi} \in \mathbb{R}^n$ denote the vector $[\phi_1, \ldots, \phi_n]$.

Since we make no structural assumptions about v, we consider estimators for computing Banzhaf values that access the function via *samples*; formally, a single sample corresponds to evaluating $v(S)$ for any S the algorithm chooses.

Banzhaf Values for Feature Attribution. As discussed in the previous section, Shapley and Banzhaf values have a number of applications in machine learning, which correspond to different choices of the set function v. For example, we can consider feature selection by defining $v(S)$ as the loss of a model trained only on the features in S. Similarly, we can consider data selection by defining $v(S)$ as the loss of a model trained only on the observations in S.

One of the most popular applications of Banzhaf values in explainable AI is *feature attribution* (also known as feature importance and feature influence). Let $\mathcal{M} : \mathbb{R}^n \to \mathbb{R}$ be a trained model which takes input $\mathbf{x} \in \mathbb{R}^n$. Our goal is to explain the predictions of the model on a particular data point $\mathbf{x}$. Given a subset of features S, define $\mathbf{x}^S$ as the observation where $\mathbf{x}_i^S = \mathbf{x}_i$ if feature $i \in S$ and, otherwise, $\mathbf{x}_i^S$ is sampled from the data distribution (possibly conditioned on the features in S). Then $v(S) = \mathbb{E}[\mathcal{M}(\mathbf{x}^S)]$, where the expectation is over the sampling of the features $i \notin S$. This expectation can typically only be estimated via sampling, meaning that we must contend with noisy access to the value function v. This settng motivates our evaluation of the robustness of Banzhaf value estimation methods in Section 4.

3 Kernel Banzhaf

The starting point of our work is a formulation of Banzhaf values in terms of a structured linear regression problem with n variables and an exponential number of rows (Hammer & Holzman, 1992). In Hammer & Holzman (1992), the authors show in Corollary 3.1 that the n parameter solution to the full regression problem exactly corresponds to the n Banzhaf values when the value function is simple (monotone and binary). For completeness, we show Theorem 3.1, which establishes the same result for *all* value functions.

Of course, solving an exponentially large regression problem is infeasible; hence, we propose Kernel Banzhaf, which approximately solves the regression problem by subsampling just a small subset of rows. We establish theoretical guarantees on how well the resulting solution approximates the Banzhaf values. We compare the Kernel Banzhaf to other Banzhaf value estimators in Section 4.

3.1 Linear Regression Formulation

Let $\mathbf{A} \in \mathbb{R}^{2^n \times n}$ be a design matrix and $\mathbf{b} \in \mathbb{R}^{2^n}$ be a target vector. We will use subsets $S \subseteq [n]$ to index the rows of $\mathbf{A}$ and entries of $\mathbf{b}$. In particular, $[\mathbf{A}]_{S,i}$ denotes the ith entry of row S. We set[1]:

$$[\mathbf{A}]_{S,i} = \begin{cases} +\frac{1}{2} & \text{if } i \in S \\ -\frac{1}{2} & \text{if } i \notin S \end{cases} \quad \text{and } \mathbf{b}_S = v(S). \tag{3}$$

For completeness, we include the following theorem that connects the regression problem with Banzhaf values. Corollary 3.1 in Hammer & Holzman (1992) shows the same result for the special case where v is monotone and binary. An extension to general value functions is straightforward.

Theorem 3.1 (Linear Regression Equivalence)**.** *Consider* $\mathbf{A}$ *and* $\mathbf{b}$ *as defined in Equation 3. Let*

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2.$$

Then $\boldsymbol{\phi} = \mathbf{x}^*$*, where* $\boldsymbol{\phi}$ *are the Banzhaf values of* v*.*

Proof of Theorem 3.1. As is standard, we have that:

$$\arg\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}.$$

We will analyze the right hand side. The (i, j) entry in $\mathbf{A}^\top \mathbf{A}$ is given by

$$[\mathbf{A}^\top \mathbf{A}]_{i,j} = \sum_{S \subseteq [n]} \left(\mathbb{1}[i \in S] - \frac{1}{2} \right) \left(\mathbb{1}[j \in S] - \frac{1}{2} \right). \tag{4}$$

If $i \neq j$, there are 2^{n-1} terms of $-\frac{1}{4}$ when $\mathbb{1}[i \in S] \neq \mathbb{1}[j \in S]$ and 2^{n-1} terms of $+\frac{1}{4}$ when $\mathbb{1}[i \in S] = \mathbb{1}[j \in S]$, hence Equation 4 is 0. If $i = j$, we have 2^n terms of $+\frac{1}{4}$ hence Equation 4 is 2^{n-2}. Together, this gives that

$$\mathbf{A}^\top \mathbf{A} = 2^{n-2} \mathbf{I}. \tag{5}$$

[1]Hammer & Holzman (1992) use a more complicated regression formulation. For ease of understanding, we simplify it slightly.

Algorithm 1 Kernel Banzhaf

Input: n: positive integer, $v : \{0,1\}^n \to \mathbb{R}$: set function, m : even number of samples such that $n < m \leq 2^n$
Output: $\hat{\boldsymbol{\phi}} \in \mathbb{R}^n$: estimated Banzhaf values
Initialize $\tilde{\mathbf{A}} \leftarrow \mathbf{0}_{m \times n}$
for $j \in \{0, 2, 4, \ldots, m-2\}$ **do**
  Sample S_j uniformly from all subsets of $[n]$
  Set $\tilde{\mathbf{A}}_j \leftarrow \mathbb{1}[S_j] - \frac{1}{2}\mathbf{1}$
  Compute $S_{j+1} = [n] \setminus S$ // Paired sampling
  Set $\tilde{\mathbf{A}}_{j+1} \leftarrow \mathbb{1}[S_{j+1}] - \frac{1}{2}\mathbf{1}$
end for
Compute $\tilde{\mathbf{b}} \leftarrow [v(S_1), \ldots, v(S_m)]$
Solve $\hat{\boldsymbol{\phi}} \leftarrow \arg\min_{\mathbf{x}} \|\tilde{\mathbf{A}}\mathbf{x} - \tilde{\mathbf{b}}\|_2$
Return: $\hat{\boldsymbol{\phi}}$

Then $(\mathbf{A}^\top \mathbf{A})^{-1} = \frac{1}{2^{n-2}}\mathbf{I}$. Continuing, we have

$$(\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top \mathbf{b} = \frac{1}{2^{n-2}}\mathbf{A}^\top \mathbf{b} = \frac{1}{2^{n-2}} \sum_{S \subseteq [n]} \left(\mathbb{1}[S] - \frac{1}{2}\mathbf{1} \right) v(S).$$

We can write the ith entry of the above as:

$$[(\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top \mathbf{b}]_i = \frac{1}{2^{n-2}} \sum_{S \subseteq [n]} \left(\mathbb{1}[i \in S] - \frac{1}{2} \right) v(S) = \frac{1}{2^{n-1}} \sum_{S \subseteq [n] \setminus \{i\}} v(S \cup \{i\}) - v(S),$$

which is exactly Equation 2. The statement follows. □

Another Regression Formulation. Ruiz et al. (1998) introduce a regression problem with the property that $\boldsymbol{\phi} = \mathbf{x}^* + c\mathbf{1}$ for $\mathbf{1} \in \mathbb{R}^n$ the all-ones vector and c an unknown number that depends on v; however, it is not clear how to estimate c efficiently to thus recover the Banzhaf values Li & Yu (2024b).

3.2 The Kernel Banzhaf Algorithm

Since $\mathbf{A}$ and $\mathbf{b}$ have 2^n rows, constructing the linear regression problem in Theorem 3.1 to calculate the Banzhaf values is computationally prohibitive. In particular, we must evaluate $v(S)$ for all 2^n subsets of $[n]$ to compute $\mathbf{b}$. Inspired by Kernel SHAP, we avoid this cost by constructing a much smaller regression problem that contains a subsample of $m \ll 2^n$ rows from $\mathbf{A}$ and $\mathbf{b}$. Let the subsampled design matrix be $\tilde{\mathbf{A}} \in \mathbb{R}^{m \times n}$ and the subsampled target vector be $\tilde{\mathbf{b}} \in \mathbb{R}^m$. The estimate we produce is

$$\hat{\boldsymbol{\phi}} = \arg\min_{\mathbf{x}} \|\tilde{\mathbf{A}}\mathbf{x} - \tilde{\mathbf{b}}\|_2^2.$$

The most direct approach to constructing $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{b}}$ would be to subsample m rows from $\mathbf{A}$ and $\mathbf{b}$ uniformly at random. We do so with a slight modification – whenever the row corresponding to a set S is sampled, we also select the row corresponding to $[n] \setminus S$. Referred to as "paired sampling", this approach has been used in the Kernel SHAP algorithm for Shapley values Covert & Lee (2020). While it has no impact on our theoretical results (unpaired sampling would give almost identical bounds) we observe an experimental improvement for Banzhaf value estimation as well. The efficacy of paired sampling has been widely observed, and recent work provides a compelling explanation Fumagalli et al. (2026a;b). Detailed pseudocode for our Kernel Banzhaf appears in Algorithm 1.

Time Complexity. Let T be the time complexity of evaluating the set function v. The total runtime of our Kernel Banzhaf method is $O(Tm + mn^2)$. In particular, it takes $O(n)$ time to sample a random set and its compliment, for a total of $O(nm)$ cost to construct $\tilde{\mathbf{A}}$. It then takes $O(Tm)$ time to construct $\tilde{\mathbf{b}}$, and finally $O(mn^2)$ time to solve the $m \times n$ least squared problem involving $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{b}}$. For most applications in

ML and explainable AI, we expect the $O(Tm)$ term to dominate. For example, even a forward pass on a shallow fully connected neural network with n features takes $O(n^2)$ time, so evaluating any attribution-based value function takes at least $O(mn^2)$ time in total. Figure 5 in Appendix D confirms that the cost of Kernel Banzhaf (and prior Monte Carlo methods) scales roughly linearly with the number of samples, m, i.e., the number of evaluations of v.

3.3 Approximation Guarantees

Despite its relative simplicity, it is possible to prove strong theoretically guarantees on the approximation quality of the Kernel Banzaf method. Such guarantees have recently been shown for regression-based Shapley evaluation using results on subsampled regression problems from randomized numerical linear algebra Musco & Witter (2025); Woodruff et al. (2014); Drineas & Mahoney (2018). In Appendix A, we use similar tools to first prove that, using a near-linear number of samples, we can solve the Banzhaf value regression formulation from Theorem 3.1 near optimally:

Theorem 3.2. *If $m = O(n \log \frac{n}{\delta} + \frac{n}{\delta\epsilon})$, Algorithm 1 produces an estimate $\hat{\boldsymbol{\phi}}$ that satisfies, with probability $1 - \delta$,*

$$\|\mathbf{A}\hat{\boldsymbol{\phi}} - \mathbf{b}\|_2^2 \leq (1+\epsilon)\|\mathbf{A}\boldsymbol{\phi} - \mathbf{b}\|_2^2. \tag{6}$$

Due to a known lower bound on the sample complexity of agnostic active linear regression Chen & Price (2019), Theorem 3.2 can only be improved in the logarithmic factor on n Chen & Price (2019). For constant ϵ and δ this is intuitive: in particular, consider a linear set function where $\mathbf{b}$ is in the span of $\mathbf{A}$. Then the right-hand side is 0 and we must recover $\boldsymbol{\phi}$ exactly. To do this, we need to observe at least n linearly independent rows of the regression problem.

As a direct consequence of Theorem 3.2, we obtain a bound on the Euclidean distances between the approximate Banzhaf values computed by our Kernel Banzhaf method and the true values. In particular, in Appendix A, we show:

Corollary 3.3. *Let $\gamma = \|\mathbf{A}\boldsymbol{\phi} - \mathbf{b}\|_2^2 / \|\mathbf{A}\boldsymbol{\phi}\|_2^2$. Any $\hat{\boldsymbol{\phi}}$ that satisfies Equation 6 also satisfies*

$$\|\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}\|_2^2 \leq \epsilon\gamma\|\boldsymbol{\phi}\|_2^2. \tag{7}$$

The reverse is also true: Equation 7 implies Equation 6.

Corollary 3.3 incurs a dependence on a parameter γ, which captures the extent to which v deviates from an additive set function (geometrically, the distance of $\mathbf{b}$ from the linear span of the rows of $\mathbf{A}$). Because there is no relaxation in the guarantee from Theorem 3.2 to Corollary 3.3—and given that the sample complexity of Theorem 3.2 is near-optimal—we argue that dependence on γ is inherent to the Kernel Banzhaf method.

Experimentally, we find this quantity is manageable: the median value of γ was approximately 100 for the datasets small enough for us to compute it. While this magnitude implies that the sample complexity must scale with γ (requiring $\epsilon < \frac{1}{100}$ and roughly $m = \frac{n}{\epsilon} = 100n$ samples for non-trivial bounds in this regime), this cost appears as a data-dependent constant factor. Consequently, the method remains computationally efficient and polynomial in n, avoiding the exponential scaling required for exact computation. In the ideal case where $\mathbf{b}$ lies exactly in the span of $\mathbf{A}$, γ vanishes ($\gamma = 0$) and we recover the exact Banzhaf values.

Comparison of Theoretical Guarantees. Basic concentration bounds can be used to prove similar guarantees for the Monte Carlo and Maximum Sample Reuse estimators (see e.g., Theorems 4.8 and 4.9 in Wang & Jia (2023)). When MC is run with $O\left(\frac{n^2}{\epsilon}\log(\frac{n}{\delta})\right)$ samples and when MSR is run with $O\left(\frac{n}{\epsilon}\log(\frac{n}{\delta})\right)$ samples, they respectively return estimates $\tilde{\boldsymbol{\phi}}$ that satisfy, with high probability,

$$\|\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}\|_2^2 \leq \epsilon \max_{S \subseteq [n]} v(S)^2.$$

That is, both prior methods obtain a bound that depends on the maximum of v, rather than $\|\boldsymbol{\phi}\|_2^2$. In general, we expect the maximum magnitude of the set function to be larger than the Banzhaf values, which measure an *average* marginal contribution. This is reflected in the superiority of Kernel Banzhaf in our experiments (see e.g., Table 1).

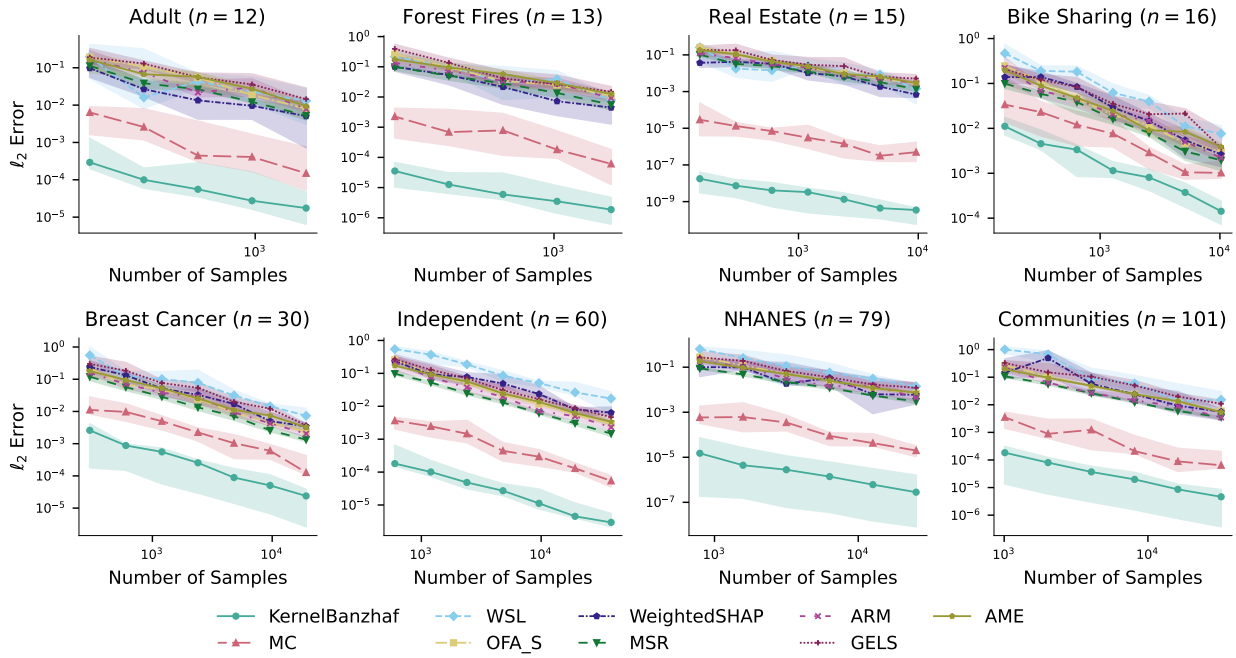

Figure 2: Plots comparing the relative squared ℓ_2-norm error (i.e., $\|\hat{\boldsymbol{\phi}}-\boldsymbol{\phi}\|_2^2/\|\boldsymbol{\phi}\|_2^2$) for each Banzhaf estimator across increasing sample sizes in eight datasets. Each point represents the median of 50 runs, with shaded areas indicating the 25th to 75th percentiles. Kernel Banzhaf outperforms the other estimators by several orders of magnitude.

3.4 Extension to Probabilistic Values

Banzhaf and Shapley values can be generalized to so-called *probabilistic values*, which have also found applications in machine learning Kwon & Zou (2022b); Li & Yu (2024b;c). In this section, we show how the regression formulation can be extended to probabilistic values.

For a weight vector $\mathbf{p} \in [0,1]^n$ such that $\sum_{\ell=0}^{n-1} \binom{n-1}{\ell} p_\ell = 1$, the *probabilistic value* is given by

$$\phi_i^{\text{prob}} = \sum_{S \subseteq [n] \setminus \{i\}} p_{|S|} [v(S \cup \{i\}) - v(S)].$$

Define the quantities a_n and b_n as follows:

$$a_n := 2 \sum_{\ell=0}^{n-1} \binom{n-1}{\ell} p_\ell^2 - \sum_{\ell=1}^{n-1} \binom{n-2}{\ell-1} (p_\ell - p_{\ell-1})^2 \tag{8}$$

$$b_n := \frac{1}{a_n} \sum_{\ell=1}^{n-1} \binom{n-2}{\ell-1} (p_\ell - p_{\ell-1})^2 . \tag{9}$$

Note that both a_n and b_n depend only on n and $\mathbf{p}$, and can be computed in $O(n)$ time. In the Banzhaf setting when $p_\ell = p_{\ell-1}$ for all ℓ, observe that $b_n = 0$.

Construct a design matrix $\mathbf{A}^{\text{prob}} \in \mathbb{R}^{2^n \times n}$ with

$$[\mathbf{A}^{\text{prob}}]_{S,i} = \begin{cases} \frac{p_{|S|-1}}{a_n} & \text{if } i \in S \\ -\frac{p_{|S|}}{a_n} & \text{if } i \notin S. \end{cases}$$

Like for Banzhaf values, let $\mathbf{b}_S^{\text{prob}} = v(S)$.

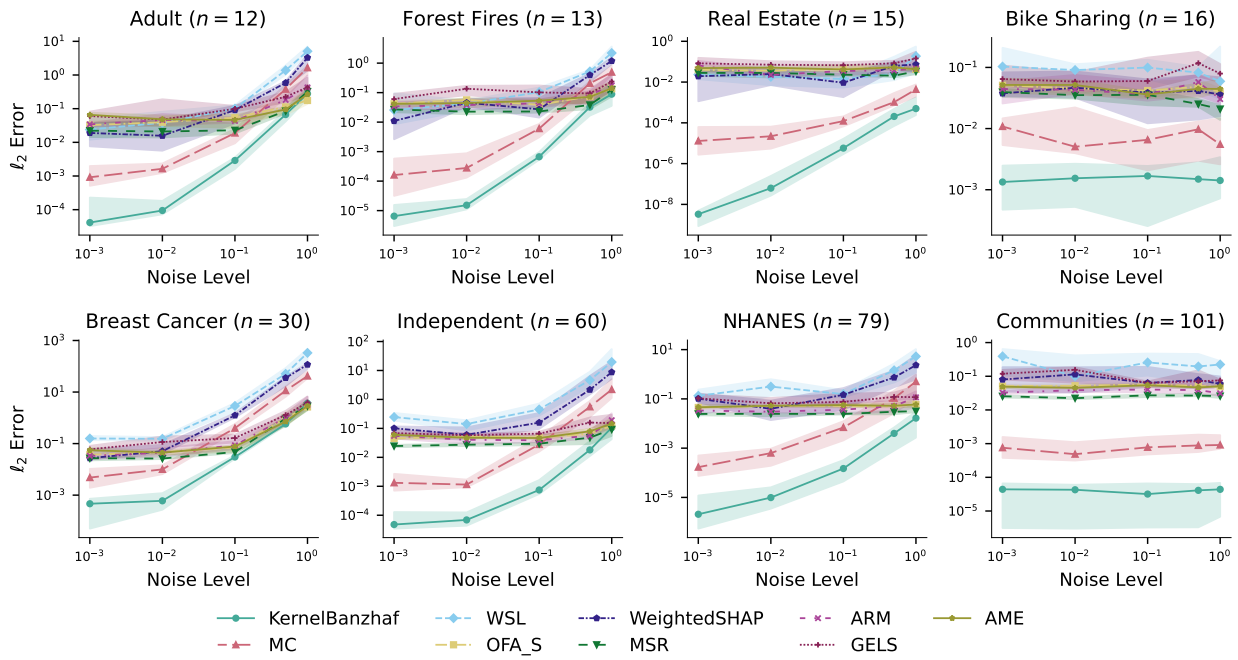

Figure 3: Plots of relative squared ℓ_2-norm error by noise level across Banzhaf estimators. For each noise level σ, the estimator observes $v(S)+x$ where $x \sim \mathcal{N}(0, \sigma)$. Kernel Banzhaf outperforms the other estimators for lower noise levels or large n.

Lemma 3.4 (Extended Regression Equivalence)**.** *Let*

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{A}^{\text{prob}}\mathbf{x} - \mathbf{b}^{\text{prob}}\|_2^2.$$

Then $\boldsymbol{\phi}^{\text{prob}} = (\mathbf{I} + b_n \mathbf{1}\mathbf{1}^\top)\mathbf{x}^*$.

We defer the proof of Lemma 3.4 to Appendix B.

Like in the Banzhaf setting, we can sample the full regression problem to produce a subsampled design matrix $\tilde{\mathbf{A}}^{\text{prob}} \in \mathbb{R}^{m \times n}$ and target vector $\tilde{\mathbf{b}}^{\text{prob}} \in \mathbb{R}^m$. Let $\tilde{\boldsymbol{\phi}}^{\text{prob}} = \arg\min_{\mathbf{x}} \|\tilde{\mathbf{A}}^{\text{prob}}\mathbf{x} - \tilde{\mathbf{b}}^{\text{prob}}\|_2^2$ be the solution to this subsampled problem. The following theorem (proven in Appendix B) gives guarantees on the accuracy of this solution.

Theorem 3.5 (Probabilistic Value Approximation)**.** *Let* $\gamma = \|\mathbf{A}^{\text{prob}}\mathbf{x}^* - \mathbf{b}^{\text{prob}}\|_2^2 / \|\mathbf{A}^{\text{prob}}\mathbf{x}^*\|_2^2$. *There is a sampling method that uses* $m = O(n \log \frac{n}{\delta} + \frac{n}{\delta\epsilon})$ *samples to compute an estimate,* $\tilde{\boldsymbol{\phi}}^{\text{prob}}$*, such that, with probability* $1 - \delta$*,*

$$\|\boldsymbol{\phi}^{\text{prob}} - \tilde{\boldsymbol{\phi}}^{\text{prob}}\|_2^2 \leq \epsilon\gamma(1 + nb_n)^3 \|\boldsymbol{\phi}^{\text{prob}}\|_2^2.$$

Note that Theorem 3.5 recovers Corollary 3.3 for Banzhaf values since $b_n = 0$. Initial experiments suggest that the dependence on b_n can limit the accuracy of this regression formulation for other probabilistic values, since we could have $b_n \gg 0$. However, we leave further exploration of the topic to future work.

4 Experiments

We conduct extensive experimental evaluations of Banzhaf estimators across eight popular datasets[2]. While Banzhaf values can be applied in many settings by changing the value function v, we focus on the canonical

[2]We will make our code publicly accessible after publication.

Table 1: The relative squared ℓ_2-norm error (i.e., $\|\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}\|_2^2 / \|\boldsymbol{\phi}\|_2^2$) of Banzhaf estimators with $m = 40n$ samples over 10 runs. We bold the lowest error for the 25% percentile, median, and 75% percentile. Across all eight datasets and every summary statistic, Kernel Banzhaf returns estimates with the lowest error, typically by an order of magnitude.

	Adult	**Forest Fires**	**Real Estate**	**Bike Sharing**	**Breast Cancer**	**Independent**	**NHANES**	**Communities**	**Mean**
KernelBanzhaf									
Mean	$\mathbf{1.50 \times 10^{-4}}$	$\mathbf{1.82 \times 10^{-5}}$	$\mathbf{3.46 \times 10^{-8}}$	$\mathbf{4.07 \times 10^{-3}}$	$\mathbf{4.44 \times 10^{-4}}$	$\mathbf{7.80 \times 10^{-5}}$	$\mathbf{8.30 \times 10^{-6}}$	$\mathbf{3.87 \times 10^{-5}}$	$\mathbf{6.01 \times 10^{-4}}$
1st Quartile	$\mathbf{3.53 \times 10^{-5}}$	$\mathbf{4.03 \times 10^{-6}}$	$\mathbf{5.32 \times 10^{-10}}$	$\mathbf{8.62 \times 10^{-4}}$	$\mathbf{5.72 \times 10^{-5}}$	$\mathbf{3.15 \times 10^{-5}}$	$\mathbf{6.10 \times 10^{-8}}$	$\mathbf{3.02 \times 10^{-6}}$	$\mathbf{1.24 \times 10^{-4}}$
2nd Quartile	$\mathbf{5.56 \times 10^{-5}}$	$\mathbf{5.95 \times 10^{-6}}$	$\mathbf{4.10 \times 10^{-9}}$	$\mathbf{3.34 \times 10^{-3}}$	$\mathbf{5.56 \times 10^{-4}}$	$\mathbf{4.84 \times 10^{-5}}$	$\mathbf{2.80 \times 10^{-6}}$	$\mathbf{3.69 \times 10^{-5}}$	$\mathbf{5.06 \times 10^{-4}}$
3rd Quartile	$\mathbf{2.82 \times 10^{-4}}$	$\mathbf{2.99 \times 10^{-5}}$	$\mathbf{1.06 \times 10^{-8}}$	$\mathbf{4.23 \times 10^{-3}}$	$\mathbf{7.20 \times 10^{-4}}$	$\mathbf{9.22 \times 10^{-5}}$	$\mathbf{1.14 \times 10^{-5}}$	$\mathbf{5.81 \times 10^{-5}}$	$\mathbf{6.78 \times 10^{-4}}$
MC									
Mean	2.49×10^{-2}	1.83×10^{-3}	1.52×10^{-5}	1.83×10^{-2}	6.71×10^{-3}	2.23×10^{-3}	8.82×10^{-4}	1.76×10^{-3}	7.08×10^{-3}
1st Quartile	3.08×10^{-4}	1.18×10^{-4}	2.33×10^{-6}	4.15×10^{-3}	2.74×10^{-3}	7.89×10^{-4}	1.52×10^{-4}	2.41×10^{-4}	1.06×10^{-3}
2nd Quartile	4.44×10^{-4}	7.84×10^{-4}	6.98×10^{-6}	1.20×10^{-2}	4.97×10^{-3}	1.46×10^{-3}	3.49×10^{-4}	1.24×10^{-3}	2.66×10^{-3}
3rd Quartile	3.69×10^{-3}	2.81×10^{-3}	9.57×10^{-6}	2.38×10^{-2}	7.19×10^{-3}	3.67×10^{-3}	7.33×10^{-4}	2.95×10^{-3}	5.61×10^{-3}
WSL									
Mean	6.38×10^{-2}	9.27×10^{-2}	8.60×10^{-2}	1.58×10^{-1}	1.21×10^{-1}	2.25×10^{-1}	2.36×10^{-1}	1.97×10^{-1}	1.47×10^{-1}
1st Quartile	2.84×10^{-2}	8.48×10^{-3}	3.78×10^{-3}	6.64×10^{-2}	5.27×10^{-2}	1.61×10^{-1}	1.44×10^{-2}	6.43×10^{-2}	4.99×10^{-2}
2nd Quartile	4.33×10^{-2}	3.10×10^{-2}	1.43×10^{-2}	1.82×10^{-1}	1.02×10^{-1}	1.87×10^{-1}	1.12×10^{-1}	9.14×10^{-2}	9.55×10^{-2}
3rd Quartile	7.02×10^{-2}	1.00×10^{-1}	1.51×10^{-1}	2.38×10^{-1}	1.69×10^{-1}	2.24×10^{-1}	3.37×10^{-1}	2.08×10^{-1}	1.87×10^{-1}
OFA_S									
Mean	5.02×10^{-2}	3.46×10^{-2}	4.74×10^{-2}	4.96×10^{-2}	4.98×10^{-2}	5.03×10^{-2}	5.18×10^{-2}	5.05×10^{-2}	4.80×10^{-2}
1st Quartile	2.53×10^{-2}	2.09×10^{-2}	4.64×10^{-2}	3.68×10^{-2}	4.13×10^{-2}	4.65×10^{-2}	4.29×10^{-2}	4.50×10^{-2}	3.81×10^{-2}
2nd Quartile	5.47×10^{-2}	3.41×10^{-2}	4.90×10^{-2}	4.62×10^{-2}	4.83×10^{-2}	5.02×10^{-2}	5.06×10^{-2}	4.81×10^{-2}	4.76×10^{-2}
3rd Quartile	7.02×10^{-2}	3.91×10^{-2}	5.25×10^{-2}	5.39×10^{-2}	5.81×10^{-2}	5.20×10^{-2}	6.22×10^{-2}	5.28×10^{-2}	5.51×10^{-2}
WeightedSHAP									
Mean	3.06×10^{-2}	3.62×10^{-2}	3.74×10^{-2}	9.58×10^{-2}	5.18×10^{-2}	7.77×10^{-2}	8.02×10^{-2}	9.01×10^{-2}	6.25×10^{-2}
1st Quartile	4.26×10^{-3}	5.86×10^{-3}	2.11×10^{-2}	6.57×10^{-2}	3.41×10^{-2}	4.39×10^{-2}	1.70×10^{-2}	2.33×10^{-2}	2.69×10^{-2}
2nd Quartile	1.34×10^{-2}	2.15×10^{-2}	3.27×10^{-2}	8.42×10^{-2}	5.15×10^{-2}	7.77×10^{-2}	1.85×10^{-2}	5.78×10^{-2}	4.47×10^{-2}
3rd Quartile	3.05×10^{-2}	5.82×10^{-2}	4.74×10^{-2}	9.76×10^{-2}	6.13×10^{-2}	1.02×10^{-1}	1.56×10^{-1}	1.37×10^{-1}	8.63×10^{-2}
MSR									
Mean	3.06×10^{-2}	2.60×10^{-2}	2.38×10^{-2}	3.56×10^{-2}	2.81×10^{-2}	2.47×10^{-2}	2.27×10^{-2}	2.65×10^{-2}	2.73×10^{-2}
1st Quartile	2.51×10^{-2}	2.00×10^{-2}	1.67×10^{-2}	2.11×10^{-2}	2.44×10^{-2}	2.25×10^{-2}	1.89×10^{-2}	2.27×10^{-2}	2.14×10^{-2}
2nd Quartile	2.66×10^{-2}	2.83×10^{-2}	2.49×10^{-2}	3.84×10^{-2}	2.85×10^{-2}	2.49×10^{-2}	2.19×10^{-2}	2.63×10^{-2}	2.75×10^{-2}
3rd Quartile	3.72×10^{-2}	3.20×10^{-2}	2.99×10^{-2}	4.82×10^{-2}	3.30×10^{-2}	2.73×10^{-2}	2.56×10^{-2}	2.97×10^{-2}	3.29×10^{-2}
ARM									
Mean	4.22×10^{-2}	3.58×10^{-2}	3.95×10^{-2}	5.16×10^{-2}	4.33×10^{-2}	4.32×10^{-2}	4.38×10^{-2}	3.08×10^{-2}	4.13×10^{-2}
1st Quartile	1.71×10^{-2}	2.36×10^{-2}	2.36×10^{-2}	2.96×10^{-2}	3.06×10^{-2}	2.82×10^{-2}	2.65×10^{-2}	2.55×10^{-2}	2.56×10^{-2}
2nd Quartile	2.19×10^{-2}	3.39×10^{-2}	3.13×10^{-2}	4.48×10^{-2}	3.79×10^{-2}	3.74×10^{-2}	3.16×10^{-2}	2.68×10^{-2}	3.32×10^{-2}
3rd Quartile	4.64×10^{-2}	4.54×10^{-2}	4.84×10^{-2}	5.13×10^{-2}	4.39×10^{-2}	4.81×10^{-2}	5.47×10^{-2}	3.18×10^{-2}	4.62×10^{-2}
GELS									
Mean	6.51×10^{-2}	6.07×10^{-2}	8.98×10^{-2}	1.47×10^{-1}	8.48×10^{-2}	7.63×10^{-2}	1.12×10^{-1}	1.05×10^{-1}	9.26×10^{-2}
1st Quartile	4.10×10^{-2}	2.71×10^{-2}	4.01×10^{-2}	6.15×10^{-2}	6.60×10^{-2}	6.37×10^{-2}	5.93×10^{-2}	6.60×10^{-2}	5.31×10^{-2}
2nd Quartile	5.53×10^{-2}	4.13×10^{-2}	5.32×10^{-2}	8.31×10^{-2}	7.63×10^{-2}	7.10×10^{-2}	6.80×10^{-2}	1.05×10^{-1}	6.91×10^{-2}
3rd Quartile	7.15×10^{-2}	7.38×10^{-2}	9.53×10^{-2}	1.33×10^{-1}	9.39×10^{-2}	7.71×10^{-2}	9.54×10^{-2}	1.37×10^{-1}	9.72×10^{-2}
AME									
Mean	5.40×10^{-2}	6.39×10^{-2}	4.21×10^{-2}	5.32×10^{-2}	5.34×10^{-2}	5.42×10^{-2}	4.82×10^{-2}	4.89×10^{-2}	5.22×10^{-2}
1st Quartile	3.12×10^{-2}	4.92×10^{-2}	3.53×10^{-2}	4.05×10^{-2}	4.24×10^{-2}	4.70×10^{-2}	4.88×10^{-2}	4.76×10^{-2}	4.28×10^{-2}
2nd Quartile	5.69×10^{-2}	5.76×10^{-2}	4.34×10^{-2}	4.77×10^{-2}	5.24×10^{-2}	5.70×10^{-2}	4.97×10^{-2}	4.89×10^{-2}	5.17×10^{-2}
3rd Quartile	7.39×10^{-2}	7.84×10^{-2}	5.07×10^{-2}	5.96×10^{-2}	6.34×10^{-2}	6.38×10^{-2}	5.01×10^{-2}	5.05×10^{-2}	6.13×10^{-2}

task of feature attribution. In addition to its popularity, focusing on feature attribution allows to exactly compute the Banzhaf values even for large $n \geq 30$, as long as the machine learning model is a tree (Karczmarz et al., 2022). For datasets with $n \leq 20$—where Banzhaf values can be exhaustively computed using Equation 2—we define the value function as the prediction made by a general neural network. We expand on the datasets and models in Appendix C.

With the exact Banzhaf values in hand, we compare the performance of Kernel Banzhaf to other Banzhaf value estimators. The primary metric we use is relative MSE i.e., $\|\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}\|_2^2 / \|\boldsymbol{\phi}\|_2^2$. In addition, we compare estimators in terms of how accurately they recover the *rank* of non-zero Banzhaf values. We measure the accuracy of the ranking using Spearman's correlation coefficient and Cayley distance (see Appendix F for details).

Baselines. We compare Kernel Banzhaf to several baselines.

- *Monte Carlo (MC)* The standard Monte Carlo estimator approximates each Banzhaf value individually. It operates by sampling each term in the summation with a probability proportional to its respective weight.

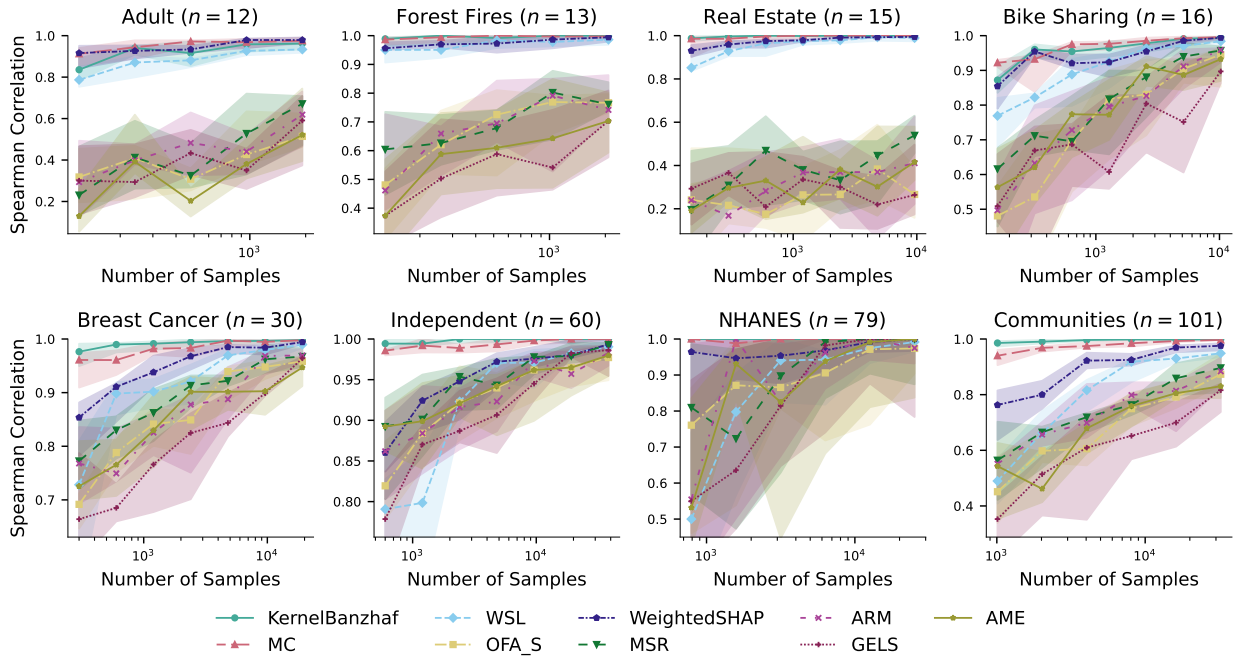

Figure 4: Comparison of non-zero Banzhaf value feature ranking recovery using Spearman correlation (higher is more accurate). Especially as the sizes of the datasets increase, Kernel Banzhaf gives the best performance.

- *Weighted Sampling Lift (WSL) (Kwon & Zou, 2022a)* This method employs a Monte Carlo approach where subsets are sampled according to Shapley weights. These samples are subsequently reweighted to produce unbiased estimates of the Banzhaf values.
- *Weighted SHAP (Kwon & Zou, 2022c)* This estimator extends permutation sampling to the Banzhaf context. Random permutations are drawn and reweighted by the Banzhaf value weights to ensure the final estimates remain unbiased.
- *Maximum Sample Reuse (MSR) Wang & Jia (2023)* MSR is derived from the observation that each Banzhaf value can be decomposed into two distinct summations: one over sets that include player i and one over sets that do not. The estimator computes each summation separately to maximize sample efficiency.
- *Approximation without Requesting Marginals (ARM) Kolpaczki et al. (2024)* A specialized variant of the MSR estimator, ARM draws half of its samples with probability $p_{|S|-1}$ and the remaining half with probability $p_{|S|}$. To mitigate the numerical instability associated with reweighting, the final estimate selectively includes the first half of samples if $i \in S$ and the second half if $i \notin S$.
- *One-Sample-Fits-All (OFA) Li & Yu (2024b)* Similar to ARM, OFA utilizes the maximum sample reuse principle but samples according to a more complex distribution to optimize estimation.
- *Generic Estimator based on Least Squares (GELS) Li & Yu (2024a)* GELS modifies the estimation process by introducing a dummy variable with a Banzhaf value of 0. Rather than fitting a linear function f iteratively, GELS utilizes the closed-form solution to the regression problem. It applies a maximum sample reuse estimator to the underlying matrix-vector multiplication, adjusting the final result by subtracting the value of the dummy variable.
- *Average Marginal Effect (AME) Lin et al. (2022)* AME utilizes an alternative regression formulation. The Banzhaf value is framed as an infinitely tall regression problem, then the estimator samples from this structure to solve an approximate version of the problem.

Sample Efficiency. Since evaluating the set function v generally dominates the estimator's time complexity (see confirmation in Appendix D), we investigate estimator error by number of samples in Figure 2. As expected, the error for all estimators decreases as the sample size increases. In terms of performance, Kernel Banzhaf gives the lowest error.

Robustness to Noise. The set functions in explainable AI tasks are often estimates of a random process e.g., the expectation of a model's prediction on a randomly sampled observation. We measure how robust each estimator is to inaccurate set function values by running experiments on the modified set function $v(S) + x$ where $x \sim \mathcal{N}(0, \sigma)$ for a given noise level σ. As shown in Figure 3, Kernel Banzhaf are more robust to this noise than the other algorithms, especially when the level of noise size is low or n is large. This suggests that Kernel Banzhaf is particularly well-suited for real-world applications where the set function is approximated.

Feature Ranking Recovery. Aside from evaluating the quantitative errors between estimated and exact Banzhaf values, we also evaluate how well each estimator recovers the rank of the non-zero Banzhaf values. (We focus only features with exact Banzhaf values that are non-zero, otherwise the rank correlation is arbitrary for these features.) Such ranking tasks, where we are interested in recovering the rank rather than quantitative value of a feature, are a principal component of comparing and selecting important features for machine learning models. Figure 4 shows the Spearman correlation coefficient between the estimated and true ranks for features with non-zero ground truth Banzhaf values. Especially for datasets with larger n, Kernel Banzhaf gives the most accurate rankings. We corroborate our findings in terms of the Cayley distance (where a lower value is better) in Figure 6 of Appendix F.

5 Conclusion and Future Work

In this work, we present Kernel Banzhaf, a regression-based approximation algorithm for Banzhaf values. Our experiments suggest that Kernel Banzhaf significantly outperforms existing methods for estimating Banzhaf values. Our theoretical analysis establishes robust approximation guarantees that depend on the size of the Banzhaf values, in contrast to Monte Carlo estimators which depend on the magnitude of the underlying set function.

One avenue for future work is to evaluate the generalization of Kernel Banzhaf to other probabilistic values. However, the poor approximation guarantee when $b_n \neq 0$ may be reflected in the generalized algorithm's performance.

References

Kevin Bache and Moshe Lichman. Uci machine learning repository, 2013.

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*, 20:105–122, 2010.

J.F. Banzhaf. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19(2):317–343, 1965.

A Charnes, B Golany, M Keane, and J Rousseau. Extremal principle solutions of games in characteristic function form: core, chebychev and shapley value generalizations. *Econometrics of planning and efficiency*, pp. 123–133, 1988.

Hugh Chen, Joseph D Janizek, Scott Lundberg, and Su-In Lee. True to the model or true to the data? *arXiv preprint arXiv:2006.16234*, 2020.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Xue Chen and Eric Price. Active regression via linear-sample sparsification. In *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pp. 663–695. PMLR, 25–28 Jun 2019.

Paulo Cortez and Aníbal de Jesus Raimundo Morais. A data mining approach to predict forest fires using meteorological data. In *Proceedings of the 13th Portuguese Conference on Artificial Intelligence*, pp. 512–523, 2007.

Ian Covert and Su-In Lee. Improving kernelshap: Practical shapley value estimation via linear regression. *arXiv preprint arXiv:2012.01536*, 2020.

Ian C. Covert, Scott Lundberg, and Su-In Lee. Understanding global feature contributions with additive importance measures. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Amit Datta, Anupam Datta, Ariel D. Procaccia, and Yair Zick. Influence in classification via cooperative game theory. In Qiang Yang and Michael J. Wooldridge (eds.), *IJCAI*, pp. 511–517. AAAI Press, 2015. ISBN 978-1-57735-738-4. URL `http://dblp.uni-trier.de/db/conf/ijcai/ijcai2015.html#DattaDPZ15`.

Petros Drineas and Michael W Mahoney. Lectures on randomized numerical linear algebra. *The Mathematics of Data*, 25(1), 2018.

Hadi Fanaee-T and Joao Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2:113–127, 2013.

Fabian Fumagalli, Landon Butler, Justin Singh Kang, Kannan Ramchandran, and R. Teal Witter. An odd estimator for shapley values, 2026a. URL `https://arxiv.org/abs/2602.01399`.

Fabian Fumagalli, R. Teal Witter, and Christopher Musco. Polyshap: Extending kernelshap with interaction-informed polynomial regression, 2026b. URL `https://arxiv.org/abs/2601.18608`.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2242–2251. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/ghorbani19c.html`.

Peter L Hammer and Ron Holzman. Approximations of pseudo-boolean functions; applications to game theory. *Zeitschrift für Operations Research*, 36(1):3–21, 1992.

Adam Karczmarz, Tomasz Michalak, Anish Mukherjee, Piotr Sankowski, and Piotr Wygocki. Improved feature importance computation for tree models based on the banzhaf value. In *Uncertainty in Artificial Intelligence*, pp. 969–979. PMLR, 2022.

Patrick Kolpaczki, Viktor Bengs, Maximilian Muschalik, and Eyke Hüllermeier. Approximating the shapley value without marginal contributions. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 38, pp. 13246–13255, 2024.

Bogdan Kulynych and Carmela Troncoso. Feature importance scores and lossless feature pruning using banzhaf power indices, 2017. URL `https://arxiv.org/abs/1711.04992`.

I Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. Problems with shapley-value-based explanations as feature importance measures. In *International conference on machine learning*, pp. 5491–5500. PMLR, 2020.

Yongchan Kwon and James Zou. Beta shapley: a unified and noise-reduced data valuation framework for machine learning. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (eds.), *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 8780–8802. PMLR, 28–30 Mar 2022a. URL `https://proceedings.mlr.press/v151/kwon22a.html`.

Yongchan Kwon and James Zou. Beta shapley: a unified and noise-reduced data valuation framework for machine learning. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (eds.), *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 8780–8802. PMLR, 28–30 Mar 2022b. URL https://proceedings.mlr.press/v151/kwon22a.html.

Yongchan Kwon and James Y Zou. Weightedshap: analyzing and improving shapley based feature attributions. *Advances in Neural Information Processing Systems*, 35:34363–34376, 2022c.

Ehud Lehrer. An axiomatization of the banzhaf value. *International Journal of Game Theory*, 17:89–99, 1988.

Weida Li and Yaoliang Yu. Faster approximation of probabilistic and distributional values via least squares. In *The Twelfth International Conference on Learning Representations*, 2024a.

Weida Li and Yaoliang Yu. One sample fits all: Approximating all probabilistic values simultaneously and efficiently. *Advances in Neural Information Processing Systems*, 2024b.

Weida Li and Yaoliang Yu. Robust data valuation with weighted banzhaf values. *Advances in Neural Information Processing Systems*, 36, 2024c.

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*, pp. 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.

Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67, 2020.

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

Christopher Musco and R. Teal Witter. Provably accurate shapley value estimation via leverage score sampling. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=wg3rBImn3O.

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*, pp. 401–411, 2021.

Luis M Ruiz, Federico Valenciano, and Jose M Zarzuelo. The family of least square values for transferable utility games. *Games and Economic Behavior*, 24(1-2):109–130, 1998.

Lloyd S Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker (eds.), *Contributions to the Theory of Games II*, pp. 307–317. Princeton University Press, Princeton, 1953.

Jakub Sliwinski, Martin Strobel, and Yair Zick. Axiomatic Characterization of Data-Driven Influence Measures for Classification. In *5th Workshop on Fairness, Accountability, and Transparency in Machine Learning (FATML), Stockholm, Sweden*, July 2018.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Jiachen T. Wang and Ruoxi Jia. Data banzhaf: A robust data valuation framework for machine learning. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 6388–6421. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/wang23e.html.

R. Teal Witter, Yurong Liu, and Christopher Musco. Regression-adjusted monte carlo estimators for shapley values and probabilistic values. *CoRR*, abs/2506.11849, 2025. doi: 10.48550/ARXIV.2506.11849. URL https://doi.org/10.48550/arXiv.2506.11849.

William Wolberg, Olvi Mangasarian, Nick Street, and W. Street. Breast cancer wisconsin (diagnostic). UCI Machine Learning Repository, 1993.

David P Woodruff et al. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014.

Yue Wu. A note on random sampling for matrix multiplication. *arXiv preprint arXiv:1811.11237*, 2018.

I-Cheng Yeh and Tzu-Kuang Hsu. Real estate valuation. UCI Machine Learning Repository, 2018.

A Kernel Banzhaf Approximation Guarantees

In this section, we prove theoretical guarantees on the performance of Kernel Banzhaf. We begin with a standard regression sampling result, the proof of which we defer to the end of this section.

Theorem A.1. *Consider a subsampling matrix* $\mathbf{S} \in \mathbb{R}^{m \times 2^n}$ *where each row contains exactly one non-zero entry, selected (possibly in a paired way) with probability proportional to the leverage scores of the corresponding row in* $\mathbf{A}$*. Define* $\tilde{\mathbf{A}} = \mathbf{SA}$ *and* $\tilde{\mathbf{b}} = \mathbf{Sb}$*. Let the solution to the subsampled problem be* $\tilde{\boldsymbol{\phi}} = \arg\min_{\mathbf{x}} \|\tilde{\mathbf{A}}\mathbf{x} - \tilde{\mathbf{b}}\|_2^2$*. If* $m = O(n \log(\frac{n}{\delta}) + \frac{n}{\delta\epsilon})$*, then*

$$\|\mathbf{A}\tilde{\boldsymbol{\phi}} - \mathbf{b}\|_2^2 \leq (1+\epsilon)\|\mathbf{A}\boldsymbol{\phi} - \mathbf{b}\|_2^2 \tag{10}$$

with probability $1 - \delta$*.*

In order to prove Theorem 3.2, we will show that uniform sampling is actually equivalent to leverage score sampling. The leverage scores of a matrix are an important statistical quantity that roughly measures the "uniqueness" of a row (Woodruff et al., 2014; Drineas & Mahoney, 2018).

The leverage scores of $\mathbf{A}$ are given by

$$\ell_S = [\mathbf{A}]_S^\top \left(\mathbf{A}^\top \mathbf{A}\right)^{-1} [\mathbf{A}]_S = \frac{1}{2^{n-2}} \left(\mathbb{1}[S] - \frac{1}{2}\right)^\top \left(\mathbb{1}[S] - \frac{1}{2}\right) = \frac{n}{2^n} \tag{11}$$

where the second equality follows by Equation 5. Notice the leverage scores are equivalent to uniform sampling and Theorem 3.2 immediately follows.

We can now prove Corollary 3.3.

Corollary 3.3. *Let* $\gamma = \|\mathbf{A}\boldsymbol{\phi} - \mathbf{b}\|_2^2 / \|\mathbf{A}\boldsymbol{\phi}\|_2^2$*. Any* $\hat{\boldsymbol{\phi}}$ *that satisfies Equation 6 also satisfies*

$$\|\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}\|_2^2 \leq \epsilon\gamma\|\boldsymbol{\phi}\|_2^2. \tag{7}$$

The reverse is also true: Equation 7 implies Equation 6.

Proof of Corollary 3.3. We have

$$\|\mathbf{A}\hat{\boldsymbol{\phi}} - \mathbf{b}\|_2^2 = \|\mathbf{A}\hat{\boldsymbol{\phi}} - \mathbf{A}\boldsymbol{\phi} + \mathbf{A}\boldsymbol{\phi} - \mathbf{b}\|_2^2 = \|\mathbf{A}\hat{\boldsymbol{\phi}} - \mathbf{A}\boldsymbol{\phi}\|_2^2 + \|\mathbf{A}\boldsymbol{\phi} - \mathbf{b}\|_2^2 \tag{12}$$

where the second equality follows because $\mathbf{A}\boldsymbol{\phi} - \mathbf{b}$ is orthogonal to any vector in the span of $\mathbf{A}$. Then, by the assumption that $\|\mathbf{A}\tilde{\boldsymbol{\phi}} - \mathbf{b}\|_2^2 \leq (1+\epsilon)\|\mathbf{A}\boldsymbol{\phi} - \mathbf{b}\|_2^2$, we have $\|\mathbf{A}\hat{\boldsymbol{\phi}} - \mathbf{A}\boldsymbol{\phi}\|_2^2 \leq \epsilon\|\mathbf{A}\boldsymbol{\phi} - \mathbf{b}\|_2^2$. By Equation 5, we have

$$\|\mathbf{A}\boldsymbol{\phi}\|_2^2 = \boldsymbol{\phi}^\top \mathbf{A}^\top \mathbf{A}\boldsymbol{\phi} = 2^{n-2}\|\boldsymbol{\phi}\|_2^2 \tag{13}$$

and, similarly, $\|\mathbf{A}(\hat{\boldsymbol{\phi}} - \boldsymbol{\phi})\ \|_2^2 = 2^{n-2}\|\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}\ \|_2^2$. Then, with the definition of γ,

$$2^{n-2}\|\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}\|_2^2 = \|\mathbf{A}(\hat{\boldsymbol{\phi}} - \boldsymbol{\phi})\|_2^2 \leq \epsilon\|\mathbf{A}\boldsymbol{\phi} - \mathbf{b}\|_2^2 = \epsilon\gamma\|\mathbf{A}\boldsymbol{\phi}\|_2^2 = 2^{n-2}\epsilon\gamma\|\boldsymbol{\phi}\|_2^2. \tag{14}$$

The statement then follows after dividing both sides by 2^{n-2}. □

Next, we prove Theorem A.1, which is a standard guarantee for leverage score sampling. However, because rows are sampled in *pairs*, we need to modify the analysis. In particular, both the spectral guarantee that the sampling matrix preserves eigenvalues and the Frobenius guarantee that the sampling matrix preserves Frobenius norm need to be reproved.

We will adopt the notation from Wu (2018). Let's consider a leverage score sampling method where rows are selected in blocks. Define Θ as a partition of blocks, each containing 2 elements with identical leverage scores in our setting. We assign a sampling probability p_i^+ to each block Θ_i, calculated as the sum of leverage scores in that block divided by the total sum of all leverage scores: $p^+ i := \frac{\sum_{k \in \Theta_i} \ell_k}{\sum_j \ell_j}$. For simplicity of notation,

suppose m is even. Let $\mathbf{S} \in \mathbb{R}^{m\times\rho}$ be a sampling matrix, initially set to $\mathbf{0}_{m\times n}$. The sampling process repeats $m/2$ times: Sample a block Θ_i with probability p_i^+. For each $k \in \Theta_i$, set the kth entry in an empty row to $\frac{1}{\sqrt{p_i^+}}$.

To analyze the solution obtained from this block-wise leverage score sampling, we will demonstrate that the sampling matrix $\mathbf{S}$ preserves both the spectral norm and the Frobenius norm.

Lemma A.2 (Spectral Approximation)**.** *Let $\mathbf{U} \in \mathbb{R}^{\rho\times n}$ be a matrix with orthonormal columns. Consider the block random sampling matrix $\mathbf{S}$ described above with rows sampled according to the leverage scores of $\mathbf{U}$. When $m = \Omega(n\log(n/\delta)/\epsilon^2)$,*

$$\|\mathbf{I} - \mathbf{U}^\top\mathbf{S}^\top\mathbf{S}\mathbf{U}\|_2 \le \epsilon \tag{15}$$

with probability $1-\delta$.

Proof of Lemma A.2. We will use the following matrix Chernoff bound (see e.g., Fact 1 in Woodruff et al. (2014)).

Fact A.3 (Matrix Chernoff)**.** *Let $\mathbf{X}_1, \ldots, \mathbf{X}_m \in \mathbb{R}^{n\times n}$ be independent samples of symmetric random matrices with $\mathbb{E}[\mathbf{X}_j] = \mathbf{0}$, $\|\mathbf{X}_j\|_2 \le \gamma$ for all j, and $\|\mathbb{E}_j[\mathbf{X}_j^2]\|_2 \le \sigma^2$. Then for any $\epsilon > 0$,*

$$\Pr\left(\left\|\frac{1}{m}\sum_{j=1}^m \mathbf{X}_j\right\|_2 \ge \epsilon\right) \le 2n\exp\left(\frac{-m\epsilon^2}{\sigma^2 + \gamma\epsilon/3}\right). \tag{16}$$

For sample $j \in [m]$, let $i(j)$ be the index of the block selected. Define

$$\mathbf{X}_j = \mathbf{I} - \frac{1}{p_{i(j)}^+}\sum_{k\in\Theta_{i(j)}} \mathbf{U}_k^\top\mathbf{U}_k \tag{17}$$

We will compute $\mathbf{E}[X_j]$, $\|\mathbf{X}_j\|_2$, and $\|\mathbb{E}[\mathbf{X}_j^2]\|_2$. First,

$$\mathbb{E}[\mathbf{X}_j] = \mathbf{I} - \sum_{i=1}^{|\Theta|} p_i^+ \frac{1}{p_i^+}\sum_{k\in\Theta_i} \mathbf{U}_k^\top\mathbf{U}_k = \mathbf{0} \tag{18}$$

where the last equality follows because Θ is a partition and $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$. Next, note that

$$\|\mathbf{X}_j\|_2 \le \|\mathbf{I}\|_2 + \frac{\sum_{k\in\Theta_{i(j)}}\|\mathbf{U}_k^\top\mathbf{U}_k\|_2}{\sum_{k\in\Theta_{i(j)}} p_k} = 1 + n\frac{\sum_{k\in\Theta_{i(j)}}\|\mathbf{U}_k\|_2^2}{\sum_{k\in\Theta_{i(j)}}\ell_k} = 1+n \tag{19}$$

where the last equality follows because $\|\mathbf{U}_k\|_2^2 = \ell_k$ since $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$. Define $\mathbf{U}_{\Theta_i} \in \mathbb{R}^{|\Theta_i|\times n}$ as the matrix with rows $\mathbf{U}_k$ for $k \in \Theta_i$. Observe that $\sum_{k\in\Theta_i}\mathbf{U}_k^\top\mathbf{U}_k = \mathbf{U}_{\Theta_i}^\top\mathbf{U}_{\Theta_i}$. Finally, note that

$$\mathbb{E}[\mathbf{X}_j^2] = \mathbf{I} - 2\sum_{i=1}^{|\Theta|} p_i^+ \frac{\sum_{k\in\Theta_i}\mathbf{U}_k^\top\mathbf{U}_k}{p_i^+} + \sum_{i=1}^{|\Theta|} p_i^+ \frac{\left(\mathbf{U}_{\Theta_i}^\top\mathbf{U}_{\Theta_i}\right)^2}{{p_i^+}^2} \tag{20}$$

$$= -\mathbf{I} + \sum_{i=1}^{|\Theta|}\frac{1}{p_i^+}\mathbf{U}_{\Theta_i}^\top\mathbf{U}_{\Theta_i}\mathbf{U}_{\Theta_i}^\top\mathbf{U}_{\Theta_i}. \tag{21}$$

Observe that entry (k,k') of $\mathbf{U}_{\Theta_i}\mathbf{U}_{\Theta_i}^\top \in \mathbb{R}^{|\Theta_i|\times|\Theta_i|}$ is $\mathbf{U}_k\mathbf{U}_{k'}^\top$. So the absolute value of each entry is $|\mathbf{U}_k\mathbf{U}_{k'}^\top| \le \|\mathbf{U}_k\|_2\|\mathbf{U}_{k'}\|_2 = \ell_k^{1/2}\ell_{k'}^{1/2}$ by Cauchy-Schwarz. Define $\ell_i^{\max} = \max_{k\in\Theta_i}\ell_k$ and $\ell_i^{\min} = \min_{k\in\Theta_i}\ell_k$. By the Gershgorin circle theorem, $\mathbf{U}_{\Theta_i}\mathbf{U}_{\Theta_i}^\top \preceq 2\ell_i^{\max}|\Theta_i|^2\mathbf{I}$. Equivalently, $\mathbf{A} \preceq \mathbf{B}$, $\mathbf{x}^\top\mathbf{A}\mathbf{x} \le \mathbf{x}^\top\mathbf{B}\mathbf{x}$ for all

$\mathbf{x}$. Consider an arbitrary $\mathbf{z}$. We have $\mathbf{z}^\top \mathbf{C}^\top \mathbf{A}\mathbf{C}\mathbf{z} \leq \mathbf{z}^\top \mathbf{C}^\top \mathbf{B}\mathbf{C}\mathbf{z}$ since $\mathbf{C}\mathbf{z}$ is some $\mathbf{x}$. It follows that $\mathbf{U}_{\Theta_i}^\top \mathbf{U}_{\Theta_i} \mathbf{U}_{\Theta_i}^\top \mathbf{U}_{\Theta_i} \preceq 2\ell_i^{\max} |\Theta_i|^2 \mathbf{U}_{\Theta_i}^\top \mathbf{U}_{\Theta_i}$. Then

$$(21) \preceq n \sum_{i=1}^{|\Theta_i|} \frac{2\ell_i^{\max} |\Theta_i|^2 \mathbf{U}_{\Theta_i}^\top \mathbf{U}_{\Theta_i}}{\sum_{k \in \Theta_i} \ell_k} \preceq n \max_i 2|\Theta_i| \frac{\ell_i^{\max}}{\ell_i^{\min}} \mathbf{I}. \tag{22}$$

Since $|\Theta_i| \leq 2$ and the leverage scores in a block are all equal, $\|\mathbb{E}[\mathbf{X}_j^2]\|_2 \leq 4n$.

Applying Fact A.3 with $m = O(n \log(n/\delta)/\epsilon^2)$ yields

$$\Pr\left(\left\| \frac{1}{m} \sum_{j=1}^{m} \mathbf{I} - \frac{1}{p_{i(j)}^+} \mathbf{U}_{\Theta_i}^\top \mathbf{U}_{\Theta_i} \right\|_2 \geq \epsilon \right) \leq \delta. \tag{23}$$

The lemma statement follows.

□

We will also show that the sampling matrix preserves the Frobenius norm.

Lemma A.4 (Frobenius Approximation)**.** *Consider the block random sampling matrix* $\mathbf{S}$ *described above with rows sampled according to the leverage scores of* $\mathbf{U} \in \mathbb{R}^{\rho \times n}$*. Let* $\mathbf{V} \in \mathbb{R}^{\rho \times n'}$*. As long as* $m \geq \frac{1}{\delta \epsilon^2}$*, then*

$$\left\| \mathbf{U}^\top \mathbf{S}^\top \mathbf{S} \mathbf{V} - \mathbf{U}^\top \mathbf{V} \right\|_F \leq \epsilon \|\mathbf{U}\|_F \|\mathbf{V}\|_F \tag{24}$$

with probability $1 - \delta$*.*

Proof of Lemma A.4. By Proposition 2.2 in Wu (2018), we have that

$$\mathbb{E}[\|\mathbf{U}^\top \mathbf{S}^\top \mathbf{S} \mathbf{V} - \mathbf{U}^\top \mathbf{V}\|_F^2] \leq \frac{1}{m} \sum_{j=1}^{|\Theta|} \frac{1}{p_{i(j)}^+} \|\mathbf{U}_{\Theta_{i(j)}}\|_F^2 \|\mathbf{V}_{\Theta_{i(j)}}\|_F^2 \tag{25}$$

where $\mathbf{U}_{\Theta_i} \in \mathbb{R}^{n \times |\Theta_i|}$ is a matrix with columns $\mathbf{U}_k$ for $k \in \Theta_i$. Because $p_k = \|\mathbf{U}_k\|_2^2 / \|\mathbf{U}\|_F^2$ by the definition of leverage scores, we have

$$\mathbb{E}[\|\mathbf{U}^\top \mathbf{S}^\top \mathbf{S} \mathbf{V} - \mathbf{U}^\top \mathbf{V}\|_F^2] \leq \frac{1}{m} \sum_{i=1}^{|\Theta|} \|\mathbf{U}\|_F^2 \frac{\sum_{k \in \Theta_i} \|\mathbf{U}_k\|_2^2}{\sum_{k \in \Theta_i} \|\mathbf{U}_k\|_2^2} \|\mathbf{V}_{\Theta_i}\|_F^2 = \frac{1}{m} \|\mathbf{U}\|_F^2 \|\mathbf{V}\|_F^2. \tag{26}$$

By Markov's inequality,

$$\Pr\left(\left\| \mathbf{U}^\top \mathbf{S}^\top \mathbf{S} \mathbf{V} - \mathbf{U}^\top \mathbf{V} \right\|_F > \epsilon \|\mathbf{U}\|_F \|\mathbf{V}\|_F \right) \leq \frac{\mathbb{E}\left[\left\| \mathbf{U}^\top \mathbf{S}^\top \mathbf{S} \mathbf{V} - \mathbf{U}^\top \mathbf{V} \right\|_F^2 \right]}{\epsilon^2 \|\mathbf{U}\|_F^2 \|\mathbf{V}\|_F^2} \leq \frac{1}{m\epsilon^2} \leq \delta \tag{27}$$

as long as $m \geq \frac{1}{\delta \epsilon^2}$. □

With Lemmas A.2 and A.4 already proved for the special paired leverage score sampling, the following analysis is standard. We include the following proof for completeness.

Proof of Theorem A.1. Observe that

$$\|\mathbf{A}\tilde{\boldsymbol{\phi}} - \mathbf{b}\|_2^2 = \|\mathbf{A}\tilde{\boldsymbol{\phi}} - \mathbf{A}\boldsymbol{\phi} + \mathbf{A}\boldsymbol{\phi} - \mathbf{b}\|_2^2 = \|\mathbf{A}\tilde{\boldsymbol{\phi}} - \mathbf{A}\boldsymbol{\phi}\|_2^2 + \|\mathbf{A}\boldsymbol{\phi} - \mathbf{b}\|_2^2 \tag{28}$$

where the second equality follows because $\mathbf{A}\boldsymbol{\phi} - \mathbf{b}$ is orthogonal to any vector in the span of $\mathbf{A}$. So to prove the theorem, it suffices to show that

$$\|\mathbf{A}\tilde{\boldsymbol{\phi}} - \mathbf{A}\boldsymbol{\phi}\|_2^2 \leq \epsilon \|\mathbf{A}\boldsymbol{\phi} - \mathbf{b}\|_2^2. \tag{29}$$

Let $\mathbf{U} \in \mathbb{R}^{\rho \times n}$ be an orthonormal matrix that spans the columns of $\mathbf{A}$. There is some $\mathbf{y}$ such that $\mathbf{U}\mathbf{y} = \mathbf{A}\boldsymbol{\phi}$ and some $\tilde{\mathbf{y}}$ such that $\mathbf{U}\tilde{\mathbf{y}} = \mathbf{A}\tilde{\boldsymbol{\phi}}$. Observe that $\|\mathbf{A}\tilde{\boldsymbol{\phi}} - \mathbf{A}\boldsymbol{\phi}\|_2 = \|\mathbf{U}\tilde{\mathbf{y}} - \mathbf{U}\mathbf{y}\|_2 = \|\tilde{\mathbf{y}} - \mathbf{y}\|_2$ where the last equality follows because $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$.

By the reverse triangle inequality and the submultiplicavity of the spectral norm, we have

$$\|\tilde{\mathbf{y}} - \mathbf{y}\|_2 \leq \|\mathbf{U}^\top\mathbf{S}^\top\mathbf{S}\mathbf{U}(\tilde{\mathbf{y}} - \mathbf{y})\|_2 + \|\mathbf{U}^\top\mathbf{S}^\top\mathbf{S}\mathbf{U}(\tilde{\mathbf{y}} - \mathbf{y}) - (\tilde{\mathbf{y}} - \mathbf{y})\|_2 \tag{30}$$

$$\leq \|\mathbf{U}^\top\mathbf{S}^\top\mathbf{S}\mathbf{U}(\tilde{\mathbf{y}} - \mathbf{y})\|_2 + \|\mathbf{U}^\top\mathbf{S}^\top\mathbf{S}\mathbf{U} - \mathbf{I}\|_2\|\tilde{\mathbf{y}} - \mathbf{y}\|_2. \tag{31}$$

Because $\mathbf{U}$ has the same leverage scores as $\mathbf{A}$ and the number of rows sampled in $\mathbf{S}$ is within a constant factor of m, we can apply Lemma A.2: With $m = O(n \log \frac{n}{\delta})$, we have $\|\mathbf{U}^\top\mathbf{S}^\top\mathbf{S}\mathbf{U} - \mathbf{I}\|_2 \leq \frac{1}{2}$ with probability $1 - \delta/2$. So, with probability $1 - \delta/2$,

$$\|\tilde{\mathbf{y}} - \mathbf{y}\|_2 \leq 2\|\mathbf{U}^\top\mathbf{S}^\top\mathbf{S}\mathbf{U}(\tilde{\mathbf{y}} - \mathbf{y})\|_2. \tag{32}$$

Then

$$\|\mathbf{U}^\top\mathbf{S}^\top\mathbf{S}\mathbf{U}(\tilde{\mathbf{y}} - \mathbf{y})\|_2 = \left\|\mathbf{U}^\top\mathbf{S}^\top\left(\mathbf{S}\mathbf{U}\tilde{\mathbf{y}} - \mathbf{S}\mathbf{b} + \mathbf{S}\mathbf{b} - \mathbf{S}\mathbf{U}\mathbf{y}\right)\right\|_2 \tag{33}$$

$$= \left\|\mathbf{U}^\top\mathbf{S}^\top\mathbf{S}\left(\mathbf{U}\mathbf{y} - \mathbf{b}\right)\right\|_2 \tag{34}$$

where the second equality follows because $\mathbf{S}\mathbf{U}\tilde{\mathbf{y}} - \mathbf{S}\mathbf{b}$ is orthogonal to any vector in the span of $\mathbf{S}\mathbf{U}$. By similar reasoning, notice that $\mathbf{U}^\top(\mathbf{U}\mathbf{y} - \mathbf{b}) = \mathbf{0}$. Then, as long as $m = O(\frac{n}{\delta\epsilon})$, we have

$$\left\|\mathbf{U}^\top\mathbf{S}^\top\mathbf{S}\left(\mathbf{U}\mathbf{y} - \mathbf{b}\right)\right\|_2 \leq \frac{\sqrt{\epsilon}}{2\sqrt{n}}\|\mathbf{U}\|_F\|\mathbf{U}\mathbf{y} - \mathbf{b}\|_2 \tag{35}$$

with probability $1 - \delta/2$ by Lemma A.4. Since $\mathbf{U}$ has orthonormal columns, $\|\mathbf{U}\|_F^2 \leq n$. Then, combining inequalities yields

$$\|\mathbf{A}\tilde{\boldsymbol{\phi}} - \mathbf{A}\boldsymbol{\phi}\|_2^2 = \|\tilde{\mathbf{y}} - \mathbf{y}\|_2^2 \leq 2\|\mathbf{U}^\top\mathbf{S}^\top\mathbf{S}\mathbf{U}(\tilde{\mathbf{y}} - \mathbf{y})\|_2 \leq \epsilon\|\mathbf{U}\mathbf{y} - \mathbf{b}\|_2 = \epsilon\|\mathbf{A}\boldsymbol{\phi} - \mathbf{b}\|_2^2 \tag{36}$$

with probability $1 - \delta$. □

B Extension to Probabilistic Values

In this section, we consider extending Kernel Banzhaf to probabilistic values. We will first establish how the optimal solution connects to the probabilistic values. Next, we will prove guarantees on the approximate solution selected via leverage score sampling.

In the remainder of this section, let $\mathbf{A} \in \mathbb{R}^{2^n \times n}$ such that

$$[\mathbf{A}]_{S,i} = \begin{cases} \frac{p_{|S|-1}}{a_n} & \text{if } i \in S \\ -\frac{p_{|S|}}{a_n} & \text{if } i \notin S \end{cases} \tag{37}$$

where $i \in [n]$ and we use $S \subseteq [n]$ as an index for ease of notation. Let $\mathbf{b} \in \mathbb{R}^{2^n}$ be the vector representation of the set function v i.e., $b_S = v(S)$.

B.1 Equivalence to Probabilistic Values

Lemma 3.4 (Extended Regression Equivalence)**.** *Let*

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{A}^{\text{prob}}\mathbf{x} - \mathbf{b}^{\text{prob}}\|_2^2.$$

Then $\boldsymbol{\phi}^{\text{prob}} = (\mathbf{I} + b_n \mathbf{1}\mathbf{1}^\top)\mathbf{x}^*$.

Proof of Lemma 3.4. Our proof strategy is to explicitly compute the optimal solution

$$\mathbf{x}^* = (\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top \mathbf{b}. \tag{38}$$

We begin with $\mathbf{A}^\top\mathbf{A}$. In the computation, we will repeatedly use the symmetry of $\mathbf{A}^\top\mathbf{A}$; namely, all the diagonal entries are the same and all the non-diagonal entries are the same. The diagonal entries for $i \in [n]$ are given by

$$a_n^2[\mathbf{A}^\top\mathbf{A}]_{i,i} = \sum_{S \subseteq [n]: i \in S} p_{|S|-1}^2 + \sum_{S \subseteq [n]: i \notin S} p_{|S|}^2 = \sum_{\ell=1}^{n} \binom{n-1}{\ell-1} p_{\ell-1}^2 + \sum_{\ell=0}^{n-1} \binom{n-1}{\ell} p_\ell^2 \tag{39}$$

$$= 2\sum_{\ell=0}^{n-1} \binom{n-1}{\ell} p_\ell^2 \tag{40}$$

where the last equality follows by a change of variable. Similarly, the off-diagonal entries for $i \neq j$ are given by

$$a_n^2[\mathbf{A}^\top\mathbf{A}]_{i,j} = \sum_{S \subseteq [n]: i,j \in S} p_{|S|-1}^2 - 2\sum_{S \subseteq [n]: i \in S, j \notin S} p_{|S|-1}p_{|S|} + \sum_{S \subseteq [n]: i,j \notin S} p_{|S|}^2 \tag{41}$$

$$= \sum_{\ell=2}^{n} \binom{n-2}{\ell-2} p_{\ell-1}^2 - 2\sum_{\ell=1}^{n-1} \binom{n-2}{\ell-1} p_{\ell-1}p_\ell + \sum_{\ell=0}^{n-2} \binom{n-2}{\ell} p_\ell^2 \tag{42}$$

$$= \sum_{\ell=1}^{n-1} \binom{n-2}{\ell-1} (p_\ell - p_{\ell-1})^2 = a_n b_n. \tag{43}$$

By the definition of a_n, we have

$$a_n = 2\sum_{\ell=0}^{n-1} \binom{n-1}{\ell} p_\ell^2 - \sum_{\ell=1}^{n-1} \binom{n-2}{\ell-1} (p_\ell - p_{\ell-1})^2 = a_n^2[\mathbf{A}^\top\mathbf{A}]_{i,i} - a_n^2[\mathbf{A}^\top\mathbf{A}]_{i,j}. \tag{44}$$

It follows that $[\mathbf{A}^\top\mathbf{A}]_{i,i} - [\mathbf{A}^\top\mathbf{A}]_{i,j} = \frac{1}{a_n}$. Then

$$\mathbf{A}^\top\mathbf{A} = ([\mathbf{A}^\top\mathbf{A}]_{i,i} - [\mathbf{A}^\top\mathbf{A}]_{i,j})\mathbf{I} + [\mathbf{A}^\top\mathbf{A}]_{i,j}\mathbf{1}\mathbf{1}^\top = \frac{1}{a_n}(\mathbf{I} + b_n\mathbf{1}\mathbf{1}^\top). \tag{45}$$

By the Sherman-Morrison formula, $(\mathbf{I} + b_n \mathbf{1}\mathbf{1}^\top)^{-1} = \mathbf{I} - \frac{b_n}{1+nb_n}\mathbf{1}\mathbf{1}^\top$. Then

$$(\mathbf{A}^\top \mathbf{A})^{-1} = a_n \left(\mathbf{I} - \frac{b_n}{1+nb_n} \mathbf{1}\mathbf{1}^\top \right). \tag{46}$$

Next, we compute $\mathbf{A}^\top \mathbf{v}$. The ith entry of this $n \times 1$ vector is given by

$$a_n [\mathbf{A}^\top \mathbf{v}]_i = \sum_{S \subseteq [n] : i \in S} p_{|S|-1} v(S) - \sum_{S \subseteq [n] : i \notin S} p_{|S|} v(S) = \sum_{S \subseteq [n] \setminus \{i\}} p_{|S|} (v(S \cup \{i\}) - v(S)) = \phi_i. \tag{47}$$

Therefore $\mathbf{A}^\top \mathbf{v} = \frac{1}{a_n} \boldsymbol{\phi}$. Finally,

$$\mathbf{x}^* = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b} = a_n \left(\mathbf{I} - \frac{b_n}{1+nb_n} \mathbf{1}\mathbf{1}^\top \right) \frac{1}{a_n} \boldsymbol{\phi}$$

and, taking the inverse,

$$\phi = \left(\mathbf{I} - \frac{b_n}{1+nb_n} \mathbf{1}\mathbf{1}^\top \right)^{-1} \mathbf{x}^* = (\mathbf{I} + b_n \mathbf{1}\mathbf{1}^\top) \mathbf{x}^*. \tag{48}$$

□

B.2 Approximation Guarantees

Theorem 3.5 (Probabilistic Value Approximation)**.** *Let $\gamma = \|\mathbf{A}^{\text{prob}} \mathbf{x}^* - \mathbf{b}^{\text{prob}}\|_2^2 / \|\mathbf{A}^{\text{prob}} \mathbf{x}^*\|_2^2$. There is a sampling method that uses $m = O(n \log \frac{n}{\delta} + \frac{n}{\delta \epsilon})$ samples to compute an estimate, $\tilde{\boldsymbol{\phi}}^{\text{prob}}$, such that, with probability $1 - \delta$,*

$$\|\boldsymbol{\phi}^{\text{prob}} - \tilde{\boldsymbol{\phi}}^{\text{prob}}\|_2^2 \leq \epsilon \gamma (1 + nb_n)^3 \|\boldsymbol{\phi}^{\text{prob}}\|_2^2.$$

We prove Theorem 3.5 by applying Theorem A.1 to the generalized regression problem. First, we compute the leverage scores of the generalized regression problem.

Lemma B.1. *Let $S \subseteq [n]$ be an index. The corresponding leverage score of the generalized regression problem is*

$$\ell_S = \frac{1}{a_n} \left[|S| p_{|S|-1}^2 + (n - |S|) p_{|S|}^2 - \frac{b_n}{1+nb_n} \left(|S| p_{|S|-1} - (n - |S|) p_{|S|} \right)^2 \right]. \tag{49}$$

Proof of Lemma B.1. Using the definition of leverage scores and Equation 46, we have

$$\ell_S = [\mathbf{A}]_S^\top (\mathbf{A}^\top \mathbf{A})^{-1} [\mathbf{A}]_S = a_n [\mathbf{A}]_S^\top \left(\mathbf{I} - \frac{b_n}{1+nb_n} \mathbf{1}\mathbf{1}^\top \right) [\mathbf{A}]_S = a_n \left(\|[\mathbf{A}]_S\|_2^2 - \frac{b_n}{1+nb_n} \|[\mathbf{A}]_S\|_1^2 \right). \tag{50}$$

By the definition of $\mathbf{A}$, the ℓ_2-norm squared is given by

$$\|[\mathbf{A}]_S\|_2^2 = \frac{1}{a_n^2} \left(\sum_{i \in S} p_{|S|-1}^2 + \sum_{i \notin S} p_{|S|}^2 \right) = \frac{1}{a_n^2} \left(|S| p_{|S|-1}^2 + (n - |S|) p_{|S|}^2 \right) \tag{51}$$

while the ℓ_1-norm squared is given by

$$\|[\mathbf{A}]_S\|_1^2 = \frac{1}{a_n^2} \left(\sum_{i \in S} p_{|S|-1} - \sum_{i \notin S} p_{|S|} \right)^2 = \frac{1}{a_n^2} \left(|S| p_{|S|-1} - (n - |S|) p_{|S|} \right)^2. \tag{52}$$

The lemma statement follows. □

Next, we prove Theorem 3.5.

Proof of Theorem 3.5. Let $\gamma = \|\mathbf{A}\mathbf{x}^* - \mathbf{b}\|_2^2 / \|\mathbf{A}\mathbf{x}^*\|_2^2$ and suppose there are $m = O(n\log(n/\delta) + n\log(1/\delta)/\epsilon)$ samples. By the standard leverage score sampling guarantee in Theorem A.1 (see e.g., the first few lines of the proof of Corollary 3.3 for the step-by-step process), we have

$$\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{A}\mathbf{x}^*\|_2^2 \leq \epsilon\|\mathbf{A}\mathbf{x}^* - \mathbf{b}\|_2^2 = \epsilon\gamma\|\mathbf{A}\mathbf{x}^*\|_2^2 \tag{53}$$

with probability $1 - \delta$. For the data matrix $\mathbf{A}$ in the unconstrained regression problem, we have

$$\|\mathbf{A}\mathbf{x}\|_2^2 = \mathbf{x}^\top\mathbf{A}^\top\mathbf{A}\mathbf{x} = \frac{1}{a_n}\mathbf{x}^\top(\mathbf{I} + b_n\mathbf{1}\mathbf{1}^\top)\mathbf{x} = \frac{1}{a_n}\left[\|\mathbf{x}\|_2^2 + b_n(\mathbf{1}^\top\mathbf{x})^2\right] \tag{54}$$

where the second equality follows by Equation 45.

Then the leverage score sampling guarantee implies

$$\frac{1}{a_n}\left[\|\mathbf{x}^* - \tilde{\mathbf{x}}\|_2^2 + b_n(\mathbf{1}^\top(\mathbf{x}^* - \tilde{\mathbf{x}}))^2\right] \leq \epsilon\gamma\frac{1}{a_n}\left[\|\mathbf{x}^*\|_2^2 + b_n(\mathbf{1}^\top\mathbf{x}^*)^2\right] \tag{55}$$

so

$$\|\mathbf{x}^* - \tilde{\mathbf{x}}\|_2^2 \leq \epsilon\gamma\left(\|\mathbf{x}^*\|_2^2 + b_n(\mathbf{1}^\top\mathbf{x}^*)^2\right) \leq \epsilon\gamma(1 + nb_n)\|\mathbf{x}^*\|_2^2 \tag{56}$$

where the last inequality follows because $(\mathbf{1}^\top\mathbf{x}^*)^2 \leq \|\mathbf{1}\|_2^2\|\mathbf{x}^*\|_2^2 = n\|\mathbf{x}^*\|_2^2$ by Cauchy-Schwarz.

Since b_n is non-negative, observe that $\|\mathbf{I} + b_n\mathbf{1}\mathbf{1}^\top\|_2 = 1 + nb_n$ and $\|(\mathbf{I} + b_n\mathbf{1}\mathbf{1}^\top)^{-1}\|_2 = 1$. It follows that

$$\|\mathbf{x}^*\|_2^2 = \|(\mathbf{I} + b_n\mathbf{1}\mathbf{1}^\top)^{-1}(\mathbf{I} + b_n\mathbf{1}\mathbf{1}^\top)\mathbf{x}^*\|_2^2 = \|(\mathbf{I} + b_n\mathbf{1}\mathbf{1}^\top)^{-1}\boldsymbol{\phi}\|_2^2 \leq \|(\mathbf{I} + b_n\mathbf{1}\mathbf{1}^\top)^{-1}\|_2^2 \cdot \|\boldsymbol{\phi}\|_2^2 = \|\boldsymbol{\phi}\|_2^2. \tag{57}$$

Then

$$\begin{aligned}
\|\boldsymbol{\phi} - \tilde{\boldsymbol{\phi}}\|_2^2 &= \|(\mathbf{I} + b_n\mathbf{1}\mathbf{1}^\top)(\mathbf{x}^* - \tilde{\mathbf{x}})\|_2^2 && (58)\\
&\leq \|\mathbf{I} + b_n\mathbf{1}\mathbf{1}^\top\|_2^2 \cdot \|\mathbf{x}^* - \tilde{\mathbf{x}}\|_2^2 && (59)\\
&\leq \epsilon\gamma(1 + nb_n)^3\|\mathbf{x}^*\|_2^2 && (60)\\
&\leq \epsilon\gamma(1 + nb_n)^3\|\boldsymbol{\phi}\|_2^2 && (61)
\end{aligned}$$

□

C Datasets and Models

Datasets

The Census Income dataset (Bache & Lichman, 2013; Covert & Lee, 2020), also known as the Adult dataset, involves predicting whether an individual's income exceeds \$50K/yr based on census data, using 14 features. The Forest Fires dataset (Cortez & Morais, 2007) uses meteorological data to predict burned area in Portugal's Montesinho Natural Park, with 13 features including temperature, humidity, and Fire Weather Index components. The Real Estate Valuation dataset (Yeh & Hsu, 2018) comprises market data from New Taipei City, Taiwan, with 15 features predicting house prices per unit area based on location, age, and proximity to transit. The Bike Sharing dataset (Fanaee-T & Gama, 2013) contains hourly rental counts from Capital Bikeshare in Washington D.C., with 16 features including weather conditions and temporal variables for demand prediction. The Breast Cancer Wisconsin (Diagnostic) dataset (Wolberg et al., 1993) includes 30 features computed from digitized images of fine needle aspirate, used for binary classification of breast masses as malignant or benign. The Independent Linear dataset is a synthetic regression dataset with 60 uncorrelated features, generated using the SHAP library (Lundberg & Lee, 2017). The NHANES dataset, with 79 features derived from the National Health and Nutrition Examination Survey (NHANES) I Epidemiologic Followup Study, models the risk of death over a 20-year follow-up period, as discussed in (Lundberg et al., 2020; Karczmarz et al., 2022). The Communities and Crime Unnormalized dataset (Bache & Lichman, 2013) aims to predict the total number of violent crimes per 100,000 population, comprising a predictive regression task with 101 features.

These datasets vary in size and column types and are predominantly utilized in previous studies for semi-value-based model explanation (Lundberg & Lee, 2017; Covert & Lee, 2020; Lundberg et al., 2020; Karczmarz et al., 2022). We primarily focus on tabular datasets because they are more thoroughly studied in this field and allow for easier acquisition of ground truth, especially in large datasets, using tree-based algorithms. Additionally, tabular datasets are prevalent in scenarios involving smaller datasets with fewer features.

Models

For the experiments involving tree set functions, we trained an XGBoost regressor model (Chen & Guestrin, 2016) with 100 trees and a maximum depth of 4. For the non-tree model experiments, we utilized a two-layer neural network equipped with a dropout layer with a rate of 0.5 to mitigate overfitting. This network was trained using a batch size of 32 and a learning rate of 0.0001, across 100 epochs. We chose this relatively simple model architecture because our primary focus is on explaining model behavior rather than maximizing its predictive accuracy.

There are generally two approaches to handling removed features in feature perturbation for general set functions, as discussed in Chen et al. (2020) and Kumar et al. (2020). Given an explicand x and a subset of features S, define $\mathbf{x}^S$ as the observation where $\mathbf{x}_i^S = \mathbf{x}_i$ if feature $i \in S$ and, otherwise, $\mathbf{x}_i^S$ is sampled from one of two distributions. The first method involves sampling from the conditional distribution of the removed features. This approach, while precise, is computationally expensive. Alternatively, the marginal distribution can be used where the observed features $\mathbf{x}_i^S$ for $i \in S$ are ignored. Due to its lower computational complexity, we adopt the latter approach.

We compute the average of the model's predictions using replacement values randomly sampled from 50 baseline points, different from the explicand. For each explicand $\mathbf{x}$, the non-selected features in $\mathbf{x}^S$ are replaced by values from baseline points, and the average of $M(\mathbf{x}^S)$ is taken to estimate the impact of marginalizing out the non-selected features. To calculate ground truth Banzhaf values, we evaluate all 2^n subsets of features in this way.

D Time Complexity

We further evaluated the computational efficiency of the Kernel Banzhaf and baseline estimators by measuring the exact time required to estimate Banzhaf values, as depicted in Figure 5. All experiments were conducted on a Lenovo SD650 with 128 GB of RAM, using only one thread for computation. While Figure 5 demonstrates that Kernel Banzhaf and MSR are fastest, we emphasize that the implementations of the estimators—especially WSL, OFA, WeightedSHAP, ARM, GELS, and AME—were not optimized for speed.

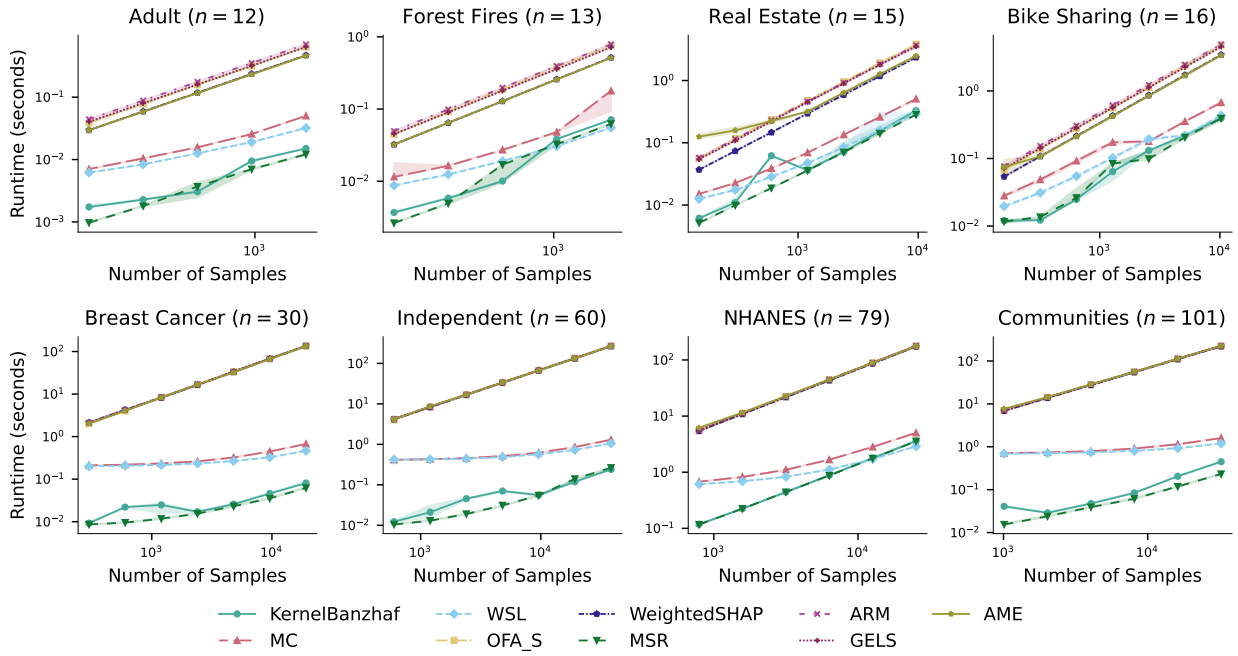

Figure 5: Computational time required for Banzhaf value estimation across varying sample sizes for eight datasets.

E Properties of Shapley Values and Banzhaf Values

Shapley values satisfy four desirable properties (Shapley, 1953):

- *Null Player* ensures that a player who does not contribute to any coalition, meaning their inclusion in any subset of players does not affect the overall outcome, is assigned a value of zero.
- *Symmetry* requires that two players who contribute equally to all possible coalitions receive the same value.
- *Linearity* requires that the Shapley value of a player in a combined game (formed by adding two games together) is equal to the sum of that player's Shapley values in the two individual games.
- *Efficiency* requires that the total value assigned to all players must sum to the value generated by the full set of players.

Banzhaf values three of the four properties, excluding *efficiency* which requires that the total value assigned to all players sum to the value generated by the full set of players (Banzhaf, 1965). Instead of the *Efficiency* property, the Banzhaf index satisfies *2-Efficiency*, which requires that the sum of the values of any two players equals the value of these two players when considered jointly in a reduced game setting (Banzhaf, 1965; Lehrer, 1988).

The necessity of the efficiency property has been debated in the context of machine learning. Sundararajan et al. (2017) suggest that the *Efficiency* property is only essential in contexts where semi-values, such as those in voting games, are interpreted numerically; Kwon & Zou (2022b) argues that the utility function in machine learning applications often does not correspond directly to monetary value, so aligning the sum of data values with total utility is unnecessary. In applications where the primary goal involves ranking features according to their importance or evaluating data, the exact numerical contribution of each feature is less critical. Both Banzhaf and Shapley values, despite their theoretical disparities, often yield the same ordering of players as shown in Karczmarz et al. (2022), which suffices for these applications. Therefore, given their efficiency and robustness properties, Banzhaf values serve as particularly effective tools in machine learning tasks (Wang & Jia, 2023).

F Feature Ranking Recovery

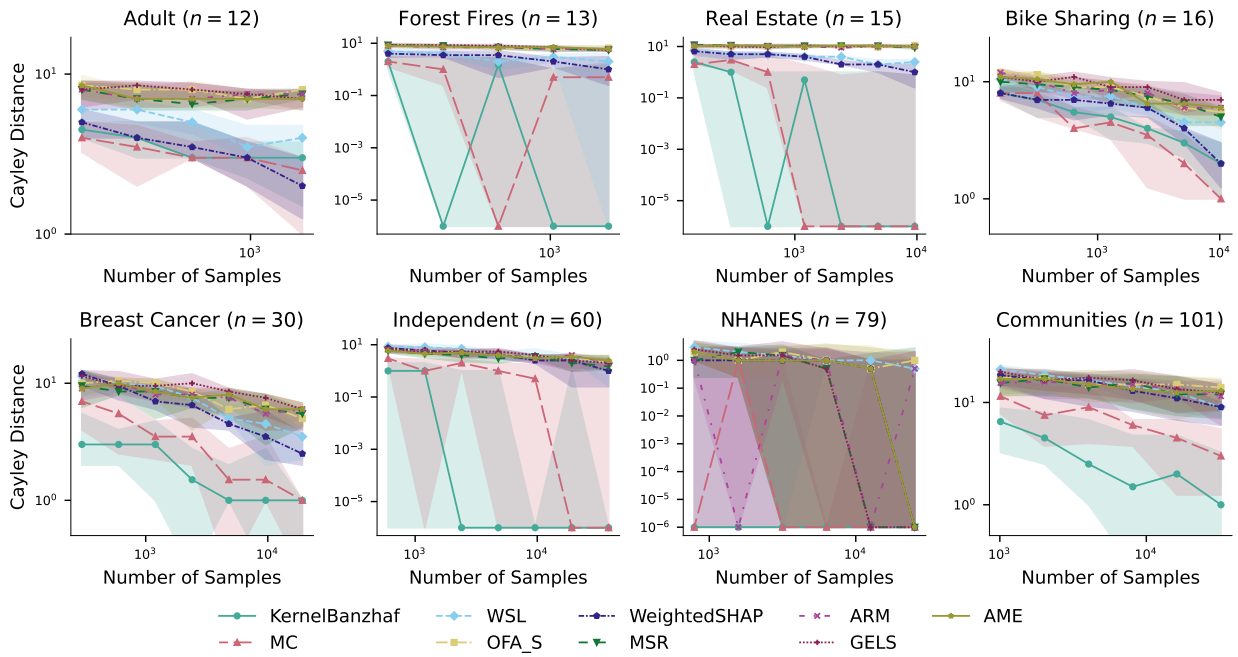

Figure 6: Comparison of non-zero Banzhaf value feature ranking recovery using Cayley distance (lower is more accurate). Kernel Banzhaf achieves the best performance in most scenarios, particularly when n is large.

Aside from evaluating the quantitative errors between estimated Banzhaf values and exact Banzhaf values, another critical metric that reveals the meaningfulness of the estimated results is how well the estimator recovers feature ranking. Feature ranking is important to feature comparison and selection, which are useful for enhancing the performance of machine learning models. Accurate feature ranking helps in identifying the most influential features, thereby facilitating more efficient and effective feature engineering and dimensionality reduction strategies.

In order to evaluate this property, we incorporate two well-known metrics: *Cayley distance* and *Spearman rank correlation*. The Cayley distance refers to the minimum number of transpositions required to transform one permutation into another. This metric provides a concrete measure of the difference between two rankings, capturing the minimal edit sequence needed, which is particularly useful in understanding the stability and reliability of feature ranking methods, and it's also adopted in Karczmarz et al. (2022). Spearman's rank correlation, ρ, on the other hand, measures the strength and direction of association between two ranked variables. Formally, it is defined as the Pearson correlation coefficient between the rank values of the variables, mathematically expressed as:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

where d_i represents the difference between the ranks of corresponding variables x_i and y_i, and n is the number of observations. This metric offers insights into how well the ranking produced by the estimator preserves the monotonic relationship compared to the exact ranking, providing a measure of ranking fidelity.

In scenarios with a large feature space, the most significant features often have a more pronounced impact on model predictions. In these cases, the overall ranking may be cluttered with a large number of features that show only minor differences in their Banzhaf values, making it difficult to distinguish among lower-ranked features effectively.

Focusing on features with non-zero ground truth Banzhaf values, therefore, targets those variables most likely to affect predictive accuracy and model stability, offering a more pragmatic evaluation of ranking recovery.

We present the results for this task in Figures 4 and 6. Our Kernel Banzhaf algorithm achieves the best performance across most datasets, demonstrating its effectiveness in identifying and ranking important features.