OR Spectrum
https://doi.org/10.1007/s00291-026-00863-0

ORIGINAL ARTICLE

Estimation of the Shapley value by ergodic sampling

Ferenc Illés[1] · **Péter Kerényi**[1]

Received: 23 July 2024 / Accepted: 13 May 2026

Abstract

The Monte Carlo approximation of the Shapley value in cooperative games was first proposed by Mann and Shapley (1960), who introduced several heuristic techniques to reduce estimation error. Since then, classical variance reduction methods from the statistical literature have been adapted to improve the efficiency of Shapley value estimators. In this paper, we demonstrate that the antithetic variates method can be successfully applied to the Monte Carlo estimation of the Shapley value. We propose an algorithm that generates pairs of negatively correlated samples, leading to a variance reduction by producing dependent observations that satisfy the Strong Law of Large Numbers, which motivates the term *ergodic sampling*. The main contribution of the paper is to establish the existence of practical, algorithmic ergodic sampling schemes for Shapley value estimation. In contrast to the heuristic approaches of Shapley and Mann, our method is data–driven: a relatively small pilot sample is used to learn an effective ergodic transformation tailored to the given game. Through numerical experiments on eight structurally diverse cooperative games, we show that the proposed approach achieves estimation accuracy comparable to other variance reduction techniques in the literature, and in several cases yields substantial variance reductions relative to independent Monte Carlo sampling.

Keywords Ergodic sampling · Antithetic variates · Variance reduction · Game theory · Shapley value

✉ Péter Kerényi
peter.kerenyi@uni-corvinus.hu

Ferenc Illés
ferenc.illes@uni-corvinus.hu

[1] Institute of Finance, Corvinus University of Budapest, Fővám tér 8., Budapest 1093, Hungary

Published online: 13 June 2026

1 Introduction

Since its introduction by Lloyd Shapley in 1953 (Shapley 1953), the *Shapley value* has become a fundamental solution concept in cooperative games with transferable utility (TU games). In such games, a group of players–called the grand coalition–can freely form subsets, or coalitions, to pursue joint outcomes, and the total value (utility) generated by any coalition is assumed to be transferable and divisible among its members. The central question addressed by the Shapley value is: *How should the total value generated by the grand coalition be fairly divided among the individual players?*

The Shapley value provides a principled answer by assigning each player their expected marginal contribution across all possible coalition formation orders. Specifically, the marginal contribution of a player to a coalition is the increase in the coalition's value when the player joins. The Shapley value takes an average of these contributions over all possible permutations of players, yielding a unique allocation rule that satisfies a compelling set of axioms–efficiency, equal treatment property, additivity, and the null player property Shapley (1953). This axiomatic characterization not only explains its intuitive appeal and fairness, but also underpins its widespread application and popularity in various domains like economics and finance (Shapley and Shubik 1969; Amihud and Barnea 1974; Csóka et al. 2009; Colini-Baldeschi et al. 2018), operations research, law (Dehez and Ferey 2013; Ferey and Dehez 2016), and, more recently, in data science and explainable machine learning (Pintér 2011; Lundberg et al. 2017; Ancona et al. 2019).

Computing the Shapley value is computationally challenging. By definition, it is obtained by averaging a player's marginal contribution over all $n!$ permutations of the players, where n is the number of players. Evaluating this expression directly therefore requires a number of operations that grows superpolynomially in n, making naïve computation impractical for large games. This computational difficulty has motivated two main lines of research on the computation of the Shapley value.

One line focuses on structured classes of games that admit compact representations via parameter sets of polynomial size (e.g., Driessen 1988; Ieong and Shoham 2005). In this setting, the computational complexity of calculating the exact Shapley value as a function of these parameters is a meaningful question. Research leads to both positive results for specific classes of games (e.g., Megiddo 1978; Granot et al. 2002; Castro et al. 2008) and hardness results (e.g., Deng and Papadimitriou 1994; Faigle and Kern 1992; Wooldridge and Dunne 2006; Bachrach and Rosenschein 2008; Csóka et al. 2022) showing intractability in general.

A second line of research considers representation–free models, in which the game is accessed through an oracle that provides the value of any coalition on request. Here, approximation algorithms, typically based on randomized sampling (Mann and Shapley 1960; Castro et al. 2009; Benati et al. 2019; Armijos-Toro et al. 2024; Touati et al. 2021; Maleki et al. 2014; Castro et al. 2017; Okhrati and Lipani 2021), are used to estimate the Shapley value. In this framework, the quality of the estimator is largely determined by its variance.

This paper contributes to the representation–free approximation line by adapting antithetic variates–a classical but previously unused tool in the Shapley context.

Antithetic sampling reduces variance by generating negatively correlated observations through a measure–preserving transformation. For the Shapley value such an approach has been missing, because the permutation space lacks a natural reflection operator and naïve transformations may increase variance.

More recently, related works have begun to investigate antithetic constructions for cooperative game values. In particular, Staudacher and Pollmann (2023) and Mitchell et al. (2022) analyze antithetic sampling schemes for Shapley-type estimators, building on ideas introduced in earlier circulated versions of our work. These approaches rely on heuristic or predefined transformations. In contrast, the present paper develops a general data-driven framework in which the transformation is learned from data, yielding an ergodic sampling scheme based on theoretical considerations.

Prior studies have shown that other statistical ideas to reduce variance like conditional sampling and stratified sampling can lead to substantial improvements in estimation accuracy (Maleki et al. 2014; Castro et al. 2017). In contrast to these approaches, which rely on independence across samples, our proposed method exploits dependence between paired permutations. This paper shows that the antithetic variates idea can be made operational in the Shapley context.

Building on the idea of antithetic variates, we propose an estimator that constructs such a transformation from data and uses it to generate correlated samples under the same computational budget as standard Monte Carlo. The resulting estimator remains unbiased and consistent, and in many classes of games it yields substantial variance reductions. Moreover, the method is complementary to existing techniques: when combined with conditioning or stratification, the hybrid estimators preserve the advantages of each approach and provide a practically useful default even when the dominant source of variance is unknown ex–ante.

To implement antithetic sampling on the permutation space, we generate an ergodic sequence of player orders based on a learned bijective transformation. Although the observations are dependent, ergodicity ensures consistency, and negative correlation between paired marginal contributions leads to variance reduction. The transformation is obtained in a pilot stage from a small sample and then used in the main estimation phase.

Extensive experiments on eight benchmark games demonstrate that the ergodic method performs well in diverse environments. At the same time, the approach is not universally beneficial: in games with highly symmetric or binary marginal–contribution structures the attainable correlation may remain non–negative, in which case the method reverts to the performance of standard Monte Carlo. The effectiveness of each individual variance reduction channel–conditioning, stratification, and ergodic sampling–varies across games, yet the hybrid procedures remain close to the best available technique in all cases. Importantly, these hybrids do not create artificial synergies: the channels remain conceptually independent and their effects are not additive, but they act as true complements by preserving the strengths of each component.

The remainder of the paper is organized as follows. Section 2 reviews cooperative games, the Shapley value, and classical variance–reduction techniques. Section 3 establishes the ergodic foundation of the ergodic estimator. Section 4 presents the proposed two–stage algorithm, and Section 5 analyzes the impact of its parameters

on variance. Section 6 reports numerical experiments and comparisons with existing methods. Section 7 concludes.

2 Preliminaries

This section introduces the notation and basic concepts used throughout the paper. We briefly review cooperative games with transferable utility and the definition of the Shapley value, and then summarize standard variance reduction techniques from the statistical literature that are relevant to our approach. All concepts presented here are well established and included solely for completeness and clarity.

2.1 Cooperative games and the Shapley value

A cooperative game with *transferable utility* (TU game) in *characteristic function form* is defined as a pair (N, v), where N is a finite set of *players*, and the characteristic function $v : 2^N \mapsto \mathbb{R}$ assigns a real number to each subset (called a *coalition*) of players, such that $v(\emptyset) = 0$. The number of players is denoted by $|N| = n$, and the number of coalitions is 2^n.

The *Shapley value* assigns to each player a unique payoff that reflects their average contribution. To define it formally, let O_N denote the set of all possible *orders* of the players in N:

$$O_N = \{o : N \longleftrightarrow \{1, 2, \ldots, n\} \text{ bijective function}\}.$$

For any order $o \in O_N$ and player $i \in N$, the value $o(i) = j$ indicates that player i is in position j in order o. The total number of such orders is $|O_N| = |N|! = n!$. For each order $o \in O_N$ and player $i \in N$, let

$$Pre_o(i) = \{j \in N \,|\, o(j) < o(i)\}$$

denote the set of players that precede i in the order o. The *marginal contribution* of player i in order o is defined as

$$M_o(i) = v(Pre_o(i) \cup \{i\}) - v\left(Pre_o(i)\right).$$

The *Shapley value* of player i is the average of its marginal contributions over all possible orders:

Definition 1 (Shapley Value)

$$Sh(i) = \frac{1}{n!} \sum_{o \in O_N} M_o(i). \tag{1}$$

The Shapley value is uniquely characterized by the following four axioms Shapley (1953), which are commonly interpreted as desirable fairness properties for allocation rules:
Efficiency: the total value of the grand coalition is fully distributed among all players:

$$\sum_{i \in N} Sh(i) = v(N).$$

Equal Treatment Property: if two players i and j contribute equally to every coalition they can join, then they receive the same payoff:

$$v\left(S \cup \{i\}\right) = v\left(S \cup \{j\}\right) \; \forall S \subseteq N \backslash \{i, j\} \Longrightarrow Sh(i) = Sh(j).$$

Null Player Property: if a player $i \in N$ adds no value to any coalition it joins, then its payoff is zero:

$$v\left(S \cup \{i\}\right) = v\left(S\right) \; \forall S \subseteq N \backslash \{i\} \Longrightarrow Sh(i) = 0.$$

Additivity: For any two games with characteristic functions *v* and *w*, the Shapley value satisfies equation

$$Sh_v(i) + Sh_w(i) = Sh_{v+w}(i).$$

An equivalent and often more computationally tractable form of Eq. (1) is obtained by grouping orders according to the set of players that precede player. Since the marginal contribution $M_o(i)$ depends only on $Pre_o(i)$, we can express the Shapley value as a sum over subsets as follows.

$$Sh(i) = \sum_{S \subseteq N \backslash \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} \Big(v\left(S \cup \{i\}\right) - v\left(S\right) \Big). \tag{2}$$

This form reduces the number of terms from $n!$ to 2^{n-1}, which still remains exponential in n, making exact computation intractable for large games.

The Shapley value admits an equivalent probabilistic interpretation. In particular, Eq. (1) can be viewed as the expected value of the marginal contribution of player i, interpreted as a random variable on a discrete (but very large) probability space induced by all possible orders in which players may join a coalition. This perspective makes explicit that the Shapley value is an expectation, and therefore it can be approximated by Monte Carlo simulation by sampling from this space and averaging the corresponding marginal contributions (for more details see, e.g., Castro et al. 2009). When the Shapley value is estimated via a randomized procedure, the defining axioms of the Shapley value are generally satisfied only in expectation rather than deterministically. Null player property holds numerically, as a player with identically zero marginal contribution yields an estimated Shapley value equal to zero. In

contrast, the efficiency axiom is not satisfied for a finite sample, but this violation can be corrected while preserving unbiasedness by simply subtracting the average Shapley value and adding $v(N)/nv$ to each component.[1] Additivity and equal treatment property cannot be satisfied simultaneously; indeed, by the uniqueness of the Shapley value, no estimation procedure can satisfy all four axioms unless it coincides with the exact Shapley value.

According to the Central Limit Theorem, the distribution of a Monte Carlo estimator of an expected value converges to normality, with its variance fully characterizing the estimator's accuracy. As a result, variance reduction is the central concern in the literature on Monte Carlo methods.

2.2 Variance reduction techniques

Variance reduction techniques are standard tools in Monte Carlo simulation that aim to reduce estimator variance without increasing computational cost. Such improvements are only possible by exploiting additional structure or information about the data–generating process beyond that used by independent sampling. Starting from an unbiased estimator, these methods modify the sampling scheme—through conditioning, stratification, or dependence between samples—while preserving unbiasedness and consistency.

Conditional sampling exploits the fact that the distribution of a player's marginal contribution depends strongly on the player's position in the order. Let $C_i(o) \in \{1, \ldots, n\}$ denote the position of player$i$ in order o. Based on the Law of Total Expectation, the Shapley value can then be written as

$$Sh_i = \sum_{k=1}^{n} \mathbb{P}(C_i = k)\mathbb{E}[M(i) \,|\, C_i = k].$$

The conditional Monte Carlo estimator draws orders independently, groups the observations according to the realized position of player i, and combines the estimated conditional means using the true probabilities $\mathbb{P}(C_i = k) = 1/n$ rather than the empirical frequencies observed in the sample.

The effectiveness of this approach relies on heterogeneity across positions: if the conditional means $\mathbb{E}[M(i) \,|\, C_i = k]$ differ substantially, conditioning can reduce variance relative to simple random sampling. This idea already appears in the early work of Mann and Shapley (1960). However, conditional sampling remains an independent–sampling technique and does not exploit potential negative dependence between observations.

Stratified sampling builds on conditional Monte Carlo by actively allocating the simulation budget across positions. Let stratum k consist of orders in which player i occupies position k.

[1] This correction is not necessarily optimal, but unbiased because of its linearity. Rescaling the vector of estimated Shapley values introduces bias.

The distinctive feature of stratification is that the subsample sizes m_k are chosen optimally according to the heterogeneity across positions. Classical theory (Neyman allocation) suggests allocating observations proportionally to the conditional standard deviations $\sigma_k = \sqrt{Var(M(i) \mid C_i = k)}$, so that more effort is devoted to strata with higher variability. Maleki et al. (2014) propose an allocation rule derived from theoretical considerations, leading to subsample sizes proportional to $k^{2/3}$. Castro et al. (2017) introduce a two–stage procedures in which a preliminary sample is used to estimate these heterogeneities and the remaining budget is distributed accordingly.

To illustrate the idea of *antithetic variates*, consider the estimation of $\mathbb{E}[f(U)]$ for a uniform variable $U \sim \mathcal{U}(0,1)$. A simple Monte Carlo estimator uses independent draws $U_1, \ldots, U_m$. Instead, one may generate paired observations $(U, 1-U)$ and average $\frac{1}{2}(f(U) + f(1-U))$. When f is monotone, the two evaluations $f(U)$ and $f(1-U)$ tend to move in opposite directions, leading to a substantial reduction in variance.

The transformation $T(U) = 1 - U$ is therefore a convenient heuristic, but its success is not universal. If the function f is not monotone, the pairing $(U, 1-U)$ may produce positive rather than negative correlation and can even increase the variance. Effective antithetic sampling thus requires a transformation adapted to the problem at hand.

Applying this principle to the Shapley value is particularly challenging. Naïve transformations on orders may increase rather than decrease variance, and negative correlation between marginal contributions is not guaranteed. The central task of this paper is therefore to construct a data–driven, measure–preserving transformation on the orders space that generates effective antithetic pairs. Section 3 develops such a transformation using an ergodic sampling scheme and analyzes the resulting estimator.

3 Theoretical background of the ergodic sampling

In this section, we recall the concept of ergodicity and introduce the construction of an ergodic seqence that forms the theoretical foundation of our proposed sampling method.

Definition 2 (*Ergodicity*) An infinite sequence of identically distributed random variables $X_1, X_2, \ldots$ with finite expected value μ is said to be ergodic,[2] if their sample mean converges to the expected value with probability 1:

$$\mathbb{P}\left(\lim_{n\to\infty} \frac{\sum X_i}{n} = \mu\right) = 1.$$

Ergodicity implies that the sample mean provides a consistent estimate of the expected value. According to the Strong Law of Large Numbers, an i.i.d. sequence is always

[2] Strictly speaking *mean ergodic* but we consider only this type of ergodicity throughout the paper.

ergodic. However, ergodicity can also arise in certain non–independent sequences, which is the key idea behind our method. To construct such an ergodic sequence, let $(\Omega_1, \mathcal{A}_1, \mathbb{P}_1)$ be a classical finite probability space, where $|\Omega_1| < \infty$, $\mathcal{A}_1 = 2^{\Omega_1}$, and the probability measure is defined as $\mathbb{P}_1(E) = \frac{|E|}{|\Omega_1|}$ for all $E \subseteq \Omega_1$. Let $X : \Omega_1 \to \mathbb{R}$ be a random variable with finite expectation $\mu = \mathbb{E}[X]$. Let $(\Omega, \mathcal{A}, \mathbb{P})$ be the countable infinite power of (Ω_1) by the usual construction of the (infinite) product of measure spaces. Elements of Ω are infinite sequences $\omega = (\omega_1, \omega_2, \dots)$, and for each $j \in \mathbb{N}$, denote the coordinate random variable $X_j(\omega) = X(\omega_j)$. By construction, the sequence X_j consists of i.i.d. copies of X. Let $t : \Omega_1 \to \Omega_1$ be a bijective transformation, and let $T : \Omega \to \Omega$ be its pointwise extension to Ω, defined by

$$T(\omega) = T(\omega_1, \omega_2, \dots) = (t(\omega_1), t(\omega_2), \dots).$$

Fix an integer parameter $K \in \mathbb{N}$ and define the following sequence of random variables:

- For all $l \geq 1$, set $Y_{1,l} = X_l$;
- For $2 \leq k \leq K$ and $l \geq 1$, define recursively:

$$Y_{k,l} = Y_{k-1,l} \circ T = Y_{1,l} \circ \underbrace{T \circ T \cdots \circ T}_{k-1} = Y_{1,l} \circ T^{k-1},$$

where $\circ$ denotes function composition and T^{k-1} is the $(k-1)$-fold composition of T.

- Define the sequence Y_j by enumerating the variables $Y_{k,l}$ in a column-wise order:

$$Y_{1,1}, Y_{2,1}, \dots Y_{K,1}, Y_{1,2}, Y_{2,2}, \dots, Y_{K,2}, Y_{1,3}, Y_{2,3}, \dots$$

Throughout the rest of the paper, we will use both single and double indices interchangeably, depending on which is more convenient. Specifically, $Y_j = Y_{l,i}$ if $j = K \cdot (i-1) + l$, for $1 \leq l \leq K$.

Our algorithm relies on the following key results:

Theorem 1 *For every integer $K \geq 2$ and bijective transformation $t : \Omega_1 \longleftrightarrow \Omega_1$, the sequence Y_j is ergodic.*

Theorem 2 *For every integer $K \geq 2$, every bijective transformation t and every positive integer m*

$$\mathbb{E}\left[\frac{\sum_{i=1}^{m} Y_j}{m}\right] = \mu$$

Theorem 3 *Even for $K = 2$ $\exists t : \Omega_1 \longleftrightarrow \Omega_1$ transformation, such that X and $X \circ t$ are negatively correlated, if the distribution of X is not degenerate.*

Theorems 1 and 2 establish two desirable statistical properties of the proposed estimator: consistency and unbiasedness. Theorem 3 asserts the existence of a transformation under which the algorithm yields an estimator with reduced variance compared to independent random sampling.[3] In the subsequent sections, we show that such a transformation can be algorithmically constructed or approximated using limited information about a TU–game. The proofs of Theorems 1 and 3 are provided in Appendix A. Since expectation is a linear operator and is unaffected by correlation between random variables, Theorem 2 follows immediately and its proof is therefore omitted.

In the subsequent sections the base space Ω_1 will be the set of orders O_N, the random variable X will correspond to the marginal contribution $M_o(i)$, and the transformation t will be a bijection on orders. Guided by Theorem 3, the remainder of the paper focuses on the practically most relevant case $K = 2$, where each draw is paired with a single antithetic counterpart. Section 4 develops a data–driven procedure for constructing such a transformation and analyzes the resulting estimator.

4 Implementation of the algorithm

4.1 Two-stage algorithm

Although the statement of Theorem 1 is abstract, generating samples from the sequence Y_j is easy in practice. We set $\Omega_1 = O_N$, the set of orders of players as defined in Sect. 2, over which we need a bijective transformation t. The algorithm has a two–stage structure: in the second stage, a sample is generated by applying transformation t, while the first stage is devoted to the algorithmic construction of this transformation. Since the sampling procedure itself is straightforward once the transformation is available, we begin by describing the second stage. We then turn to the first stage, which addresses the nontrivial task of identifying a transformation that induces the required sampling structure.

Stage 2

As discussed in the previous section, we present the algorithm for the case of $K = 2$, which simplifies the notation. A high–level heuristic description is presented below.

1. Let m_2 be a positive integer.
2. Generate an initial random player order $o_1 \sim \text{Uniform}(O_N)$.
3. Transform it into a new order $o_2 = t(o_1) \in O_N$
4. Compute and store the marginal contributions of player i for both orders $Y_1 = M_{o_1}(i)$ and $Y_2 = M_{o_2}(i)$.

[3] The variance reduction follows from the presence of negative correlation between sampled terms, as opposed to the independent sampling case.

5. Repeat the process m_2 times.[4]
6. Compute the mean of the entire sample.

A pseudo–code of this algorithm *ErgodicSampling* is presented in Appendix B. It uses subroutines transformation t and the marginal contribution function over player set $\{1, 2, \ldots, n\}$, which represents the game as an *oracle* as discussed in Sect. 1.

Stage 1

Based on Theorems 1 and 2 the sampling algorithm *ErgodicSampling* always provides a consistent and unbiased estimator. However, its practical utility crucially relies on transformation t. Since the variance of the sample mean decreases under negative correlation, *ErgodicSampling* improves estimation accuracy if and only if t induces negative correlation between the generated samples. In this section, we provide a general algorithm to construct a suitable transformation that is applicable across game classes. While tailored heuristics can significantly improve the performance, our goal here is to test the theoretical feasibility of ergodic sampling through a generic procedure. Developing game–specific transformations is a task for future research and practical implementation.

The algorithm, called *GreedyK2*, attempts to find pairs $(o_1, o_2) \in O_N \times O_N$ of player orders such that the marginal contributions $M_{o_1}(i)$ and $M_{o_2}(i)$ for player i are as negatively correlated as possible. Let

$$S_n = \{\pi : \{1, 2, \ldots, n\} \longleftrightarrow \{1, 2, \ldots, n\} \text{ bijection } \}$$

denote the symmetric group of order n. For any $o \in O_N$ and $\pi \in S_n$ $(\pi \circ o) \in O_N$. For a fixed permutation $\pi \in S_n$, the transformation $t_\pi : O_N \mapsto O_N$ for which $t_\pi(o) = \pi \circ o$ satisfies the assumptions of Theorem 1.

A permutation $\pi \in S_n$ of order 2 (meaning that $\pi^2 = id \in S_n$) is a product of disjoint transpositions. Denote by $\langle a, b\rangle \in S_n$ the transposition swapping elements $a, b \in \{1, 2, \ldots, n\}$. The algorithm proceeds as follows:

1. Choose a suitable sample size $m_1 \in \mathbb{N}$.
2. Generate m_1 i.i.d. player orders $o_1, o_2, \ldots o_{m_1}$, with replacement. Let $\mathcal{O}$ denote this sample of orders.
3. For each $a \leq b \in \{1, 2, \ldots, n\}$, define $\mathcal{O}_{a,b} = \{\langle a, b\rangle \circ o_j \,|\, o_j \in \mathcal{O}\}$.
4. For each such pair, compute the empirical covariance $c_{a,b}$ between the samples $\{M_o(i) \,|\, o \in \mathcal{O}\}$ and $\{M_o(i) \,|\, o \in \mathcal{O}_{a,b}\}$
5. Sort all $c_{a,b}$ in ascending order: $c_{a_1,b_1} \leq c_{a_1,b_2} \leq \cdots \leq c_{a_n,b_n}$. The length of this sequence is $\binom{n}{2}$.
6. Initialize an empty list E_{op}. Iteratively select the lowest covariance pair (a_k, b_k) that does not overlap with already chosen indices, and add it to E_{op}.
7. Define $\pi = \prod_{(i,j)\in E_{op}} \langle i, j\rangle$, and let $t_\pi(o) = \pi \circ o$ for all $o \in O_N$.

[4] m_2 denotes half of the sample size in the second stage of the algorithm.

This procedure is equivalent to building a complete weighted graph on n nodes, with edge weights $c_{a,b}$, and finding a greedy minimum–weight matching. Step 6 describes the greedy algorithm: We sort the edges in ascending order by their weights and always choose the next independent edge. The last three steps of the algorithm can be rephrased as follows. Let C be a matrix of the covariances calculated in the first four steps, containing the variance of the sequence $M_o(i)$ in the diagonal. Let a_1 and b_1 be the row and the column of the minimum of the matrix. Record the pair (a_1, b_1) and delete rows and columns a_1 and b_1 from the matrix (both rows and columns, because the matrix is symmetric). Now apply the same step to the smaller matrix until the matrix has no rows and columns to get a sequence of pairs (a_j, b_j). Define t_π as the transformation that swaps numbers a_j and b_j. The pseudo–code of this algorithm is presented in Appendix B.

The role of the algorithm parameters m_1 and m_2 is examined in the subsequent sections: Sect. 4.3 discusses their implications for computational complexity, while Sect. 5 analyzes how the choice of m_1 affects the variance of the final estimator. Before that, we illustrate the algorithm on a simple example.

4.2 Illustration of the ergodic sampling algorithm

The final algorithm *OptK2Shapley* is simply a combination of Stage 1, algorithm *GreedyK2* and Stage 2, *ErgodicSampling*, which is straightforward: first we construct a suitable transformation t in Stage 1 and then use it in the sampling process in Stage 2. As the entire algorithm is very nontrivial, let us illustrate how it works using a numeric example.

Consider a four–player game with players *A*, *B*, *C*, and *D*. The value of any three–player coalition is 2, the grand coalition is worth 12, and all other coalitions have a value of 0. Our goal is to estimate the Shapley value of player *A*.

Stage 1

Table 1 Illustration of Algorithm OptK2Shapley

Order_0	M_0	$\text{Order}_{3,4}$	$M_{3,4}$	$M_{2,4}$	$M_{2,3}$	$M_{1,2}$	$M_{1,3}$	$M_{1,4}$	M_t
CBAD	2	CBDA	10	2	0	2	0	2	0
DABC	0	DACB	0	10	2	0	0	0	10
BDAC	2	BDCA	10	2	0	2	0	2	0
BACD	0	BADC	0	10	2	0	0	0	10
ABDC	0	ABCD	0	0	0	0	2	10	2
BADC	0	BACD	0	10	2	0	0	0	10
DCBA	10	DCAB	0	0	10	10	10	0	0
BCAD	2	BCDA	10	2	0	2	0	2	0
ABCD	0	ABDC	0	0	0	0	2	10	2
DBCA	10	DBAC	2	0	10	10	10	0	0

Column M_0 contains the marginal contributions of player A in order in column 1. Column $M_{j,l}$ contains marginal contributions of orders if we swap players in positions j and . Column M_t is the marginal contributions of orders after swapping all pairs found by the algorithm

1. The algorithm begins by randomly generating $m_1 = 10$ player orders. The first column of Table 1 lists these initial orders, while the second column displays the corresponding marginal contributions of player A.
2. Next, the algorithm generates a new set of orders by swapping the third and fourth players in each of the original orders. These new orders and their corresponding marginal contributions are shown in the third and fourth columns of the Table 1.
3. The algorithm proceeds similarly by performing all possible pairwise swaps, i.e., for each pair (i, j) with $1 \leq i \leq j \leq n$, it swaps the ith and jth players in the original orders, generating $\binom{n}{2}$ new order sets. For the sake of simplicity, we do not list all of these in Table 1, but their corresponding marginal contributions are included in subsequent columns (from the fifth onward).[5]

4. Once all marginal contributions are computed, the algorithm constructs a graph with nodes labeled $\{1, 2, \ldots, n\}$, where edge weights represent the covariances between the original column M_0 and each transformed column $M_{i,j}$ in Table 1. This graph is visualized in Fig. 1.
5. The algorithm then finds a minimum–weight matching in the graph using a greedy procedure. In this example, it first selects the pair (2, 4), then (1, 3). The resulting transformation is:

$$t(1) = 3, \quad t(2) = 4, \quad t(3) = 1, \quad t(4) = 2,$$

which is the output of algorithm GreedyK2. Notably, this transformation tends to flip positive marginal contributions to zero and vice versa, creating a negative correlation between the marginal contributions on the original and on the transformed samples. This effect is evident in Table 1, where the final column lists the transformed marginal contributions. The correlation between the original column M_0 and the transformed column M_t is -53%, indicating effective variance

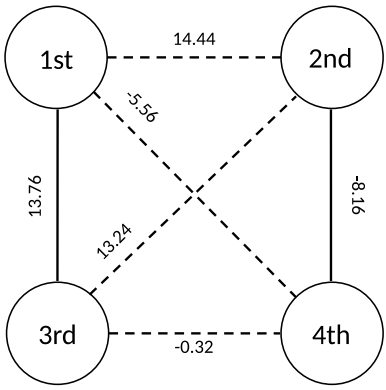

Fig. 1 Weighted graph for Algorithm GreedyK2. Edges of the matching (solid lines) indicate that the algorithm suggests swapping the second and the fourth players as well as the first and third ones to generate correlated samples

[5] For instance, to compute the fourth entry in column $M_{2,3}$, we take the fourth original order, BACD, swap the second and third players to obtain BCAD, and calculate the marginal contribution of player A to the coalition {B, C}

reduction. This example also illustrates that the transformation is player–specific; a different player would yield a different matching.*Stage 2*

1. The algorithm generates m_2 new random player orders independently and applies the transformation to each of them to obtain another m_2 orders. For example, starting from order ACDB, the marginal contribution of player A is 0. Applying the transformation (swapping the second and fourth players, and then the first and third), we obtain order DBAC, where the marginal contribution of A is 2.
2. Finally the algorithm calculates the marginal contribution of player i for all of the $2m_2$ orders and returns their mean.

4.3 Computational complexity of the ergodic sampling algorithm

Algorithm *OptK2Shapley* consists of two main stages. First, it uses a portion of the available computational budget to constract a transformation that produces negatively correlated sample pairs (pilot phase). Then, in the second stage, it generates those correlated samples and uses them to estimate the Shapley value (ergodic sampling phase). Let m_1 denote the size of sample used in the first stage, let $2m_2$ denote the size of the pair of samples generated in the second stage and let G_n denote the complexity of a game with n players, that is, the evaluation of the characteristic function. The number of elementary operations of the entire algorithm *OptK2Shapley* is

$$m_1 \binom{n}{2} G_n + 2m_2 G_n \approx m_1 \frac{n^2}{2} G_n + 2m_2 G_n, \tag{3}$$

if G_n is at least linear in n. The complexity expression 3 is based on measuring the computational cost in terms of oracle evaluations. This is appropriate when the cost of evaluating the game G_n is at least linear in the number of players, in which case oracle evaluations dominate the overall runtime and the number of elementary operations is of the same order. This condition is satisfied for all games considered in our experiments (see Sect. 6.1), and is generally a realistic assumption in practical cooperative–game applications.

Under this assumption, comparing methods under an identical number of oracle evaluations yields a fair assessment of computational complexity. When oracle evaluations are relatively cheap, however, algorithmic overhead (e.g. sorting, matrix operations, generating random values, etc.) may become non-negligible, and alternative methods with lower overhead could exploit a larger effective sample size under a fixed time budget. We therefore interpret our comparisons within the standard fixed-sample-budget framework.

Note that swapping the ath and bth players in an order o only affects player i's marginal contribution if the swap results in a new coalition being formed—that is, if $a \leq o(i) \leq b$ or $b \leq o(i) \leq a$. If both a and b are either strictly less than or strictly greater than $o(i)$, then $M_o(i)$ remains unchanged, and re–evaluating the characteristic function is unnecessary. This change in the marginal contribution occurs only with

probability approximately $\frac{1}{3}$, allowing us to effectively divide the expected running time of the Stage 1, *GreedyK2* by a factor of 3 and obtain

$$m_1 \frac{n^2}{6} G_n + 2m_2 G_n$$

for the exptected running time of the entire algorithm *OptK2Shapley*. We choose m_1 and m_2 so that the total computational complexity of the algorithm matches that of generating an independent sample of size m, which leads to equation

$$mG_n = 2m_2 G_n + \frac{n^2}{6} m_1 G_n \Longrightarrow m = 2m_2 + \frac{n^2}{6} m_1.$$

If Stage 1 tries to use a sample of size $m_1 > 6m/n^2$, when m_2 becomes negative, then the complexity of Stage 1 alone is higher than the complexity of the benchmark (a simple random sample of size m), therefore

$$m_1 \leq 6\, m/n^2$$

is a global upper bound for the sample size we can use in Stage 1, so that a positive sample size remains for Stage 2.

As a conclusion, for any positive integer m, if algorithm *OptK2Shapley* generates a sample of size

$$m_1 \leq 6m/n^2 \tag{4}$$

in Stage 1, and

$$2m_2 = m - \frac{n^2}{6} m_1 \tag{5}$$

in Stage 2, then the total expected running time—including transformation construction—matches the complexity of simple random sampling with sample size m.

5 Parameter choice and the performance of the ergodic sampling

Building on the algorithmic framework developed in Sect. 4, this section analyzes how the allocation of the computational budget between the pilot phase (m_1) and the main ergodic sampling phase (m_2) determines the variance of the ergodic estimator.

For a given m, the variance of the simple random sampling estimator reads as

$$\sigma_R = \frac{\sigma}{\sqrt{m}}, \tag{6}$$

where σ is the theoretical standard deviation of the marginal contribution as a random variable.

As we saw in the Sect. 4.3, for given m and m_1, the computational budget of the ergodic sampling and simple random sampling will be equal if $2m_2 = m - \frac{n^2}{6}m_1$. The algorithm *OptK2Shapley* generates a sample of independent and identically joint distributed pairs

$$(Y_{1,1}, Y_{2,1}), (Y_{1,2}, Y_{2,2}), \ldots, (Y_{1,m_2}, Y_{2,m_2})$$

where each two–dimensional pair $(Y_{1,j}, Y_{2,j})$ consists of correlated variables. In Stage 1, the algorithm *GreedyK2* seeks to minimize the correlation between these variables:

$$\rho = \operatorname{corr}(Y_{1,j}, Y_{2,j}).$$

After Stage 1, the transformation t and therefore the correlation ρ are fixed. Then the variance of the overall ergodic estimator, calculated in Stage 2 can be written as a function of the learned correlation, specifically,

$$\begin{aligned}
&\operatorname{Var}\left(\frac{Y_{1,1,} + Y_{2,1} + Y_{1,2} + Y_{2,2} + \cdots + Y_{1,m_2} + Y_{2,m_2}}{2m_2}\right) \\
&= \operatorname{Var}\left(\frac{\frac{Y_{1,1}+Y_{2,1}}{2} + \frac{Y_{1,2}+Y_{2,2}}{2} + \cdots + \frac{Y_{1,m_2}+Y_{2,m_2}}{2}}{m_2}\right) \\
&= \frac{1}{m_2^2} m_2 \operatorname{Var}\left(\frac{Y_{1,1} + Y_{2,1}}{2}\right) \\
&= \frac{1}{m_2}\frac{1}{4}\left(\operatorname{Var}(Y_{1,1}) + \operatorname{Var}(Y_{2,1}) + 2\operatorname{cov}(Y_{1,1}, Y_{2,1})\right) \\
&= \frac{1}{4m_2}\left(\sigma^2 + \sigma^2 + 2\sigma\sigma\rho\right) = \frac{1}{4m_2}\left(2\sigma^2(1+\rho)\right) = \frac{\sigma^2}{2m_2}(1+\rho).
\end{aligned}$$

Taking the square root, we find that the standard deviation of the ergodic estimator,

$$\sigma_E = \frac{\sigma}{\sqrt{2m_2}}\sqrt{(1+\rho)}$$

is analogous in form to the standard deviation of simple random sampling (Eq. 6), but with an adjustment factor $\sqrt{1+\rho}$ and a smaller effective sample size ($2m_2$ vs.m), since m_1 part of the computation budget is allocated to construct the ergodic transformation.

We measure the performance as the ratio of the standard deviations of ergodic and simple random estimators using the same computational budget, and refer to that as the *improvement ratio*, which is given by

$$\frac{\sigma_E}{\sigma_R} = \frac{\frac{\sigma}{\sqrt{2m_2}}\sqrt{1+\rho}}{\frac{\sigma}{\sqrt{m}}} = \sqrt{\frac{m}{2m_2}(1+\rho)}. \tag{7}$$

If m_1 is fixed and $m \to \infty$, then $\frac{m}{2m_2} \to 1$, the asymptotic improvement ratio becomes

$$\lim_{m\to\infty} \frac{\sigma_E}{\sigma_R} = \sqrt{1+\rho}. \tag{8}$$

Ergodic sampling outperforms simple random sampling (i.e., has a lower standard deviation) if and only if the improvement ratio is less than 1:

$$\sqrt{\frac{m}{2m_2}(1+\rho)} \leq 1 \iff m(1+\rho) \leq 2m_2 \iff m_1 \leq -6\rho\frac{m}{n^2}. \tag{9}$$

Condition 9 does not imply a practical rule for parameter choice—since ρ, as the correlation of random variables, is unknown—but it does characterize the theoretical boundary under which ergodic sampling can, in principle, dominate simple random sampling.

We note that the correlation coefficient ρ can depend on m_1. Since algorithm *GreedyK2* is more likely to identify (or *learn*) an accurate transformation when more data are available, $\rho(m_1)$ is expected to decrease as a function of m_1, although non–monotone realizations may occur for particular samples. It is therefore reasonable to assume that the improvement ratio is increasing in ρ (that is, more negative values of ρ decrease it), decreasing in m_2 (which itself decreases with m_1), and U–shaped as a function of m_1. This reflects a trade–off: increasing m_1 improves the learned transformation and reduces $\rho(m_1)$, but it also reduces the remaining sample size m_2 available for Stage 2. For small values of m_1, m_2 remains large, but the correlation may still be close to 1. For sufficiently large m_1, the correlation may become strongly negative, albeit at the cost of increased resource consumption in Stage 1, which in turn reduces m_2. This trade–off vanishes as $m \to \infty$, since further increasing m_1 no longer leads to a substantial improvement in $\rho(m_1)$.

Figure 2 illustrates the mechanism behind the improvement ratio (Eq. 7) under two hypothetical scenarios for the behavior of the learned correlation as a function of m_1 . The analysis should be interpreted as a stylized description of the average tendency of the learning process rather than as a deterministic property guaranteed for every sample path. In both scenarios $\rho(m_1)$ is expected to decrease and converge, although finite–sample estimates may display non–monotone fluctuations. The scenarios differ only in their convergence limits, and the assumed shapes are in line with the empirical patterns reported in Sect. 6.

Panel 2a on the left corresponds to the case in which $\rho(m_1)$ converges to 0 as the pilot size m_1 increases. The upper–left subplot shows this assumed correlation curve, together with a horizontal solid gray line marking the level $\rho = 0$. Since the curve never crosses this line, negative correlation cannot be achieved, and the ergodic estimator cannot outperform the simple random sampling. The lower–left subplot

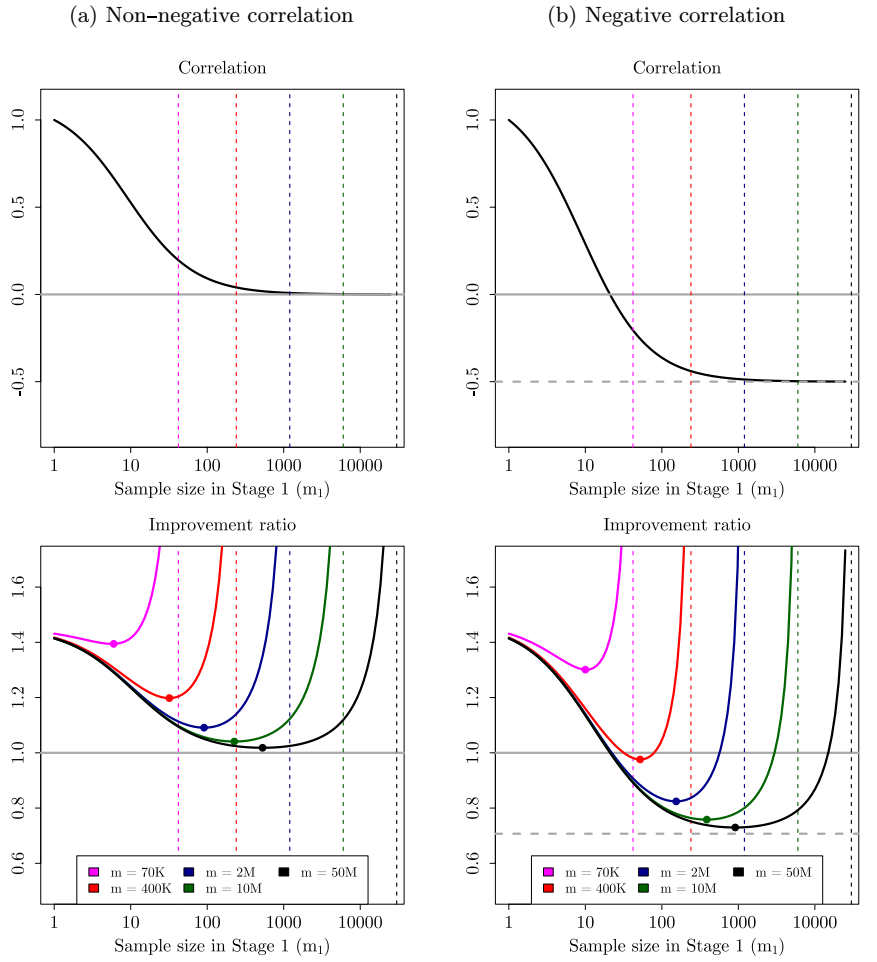

Fig. 2 Stylized illustration of the improvement ratio σ_E/σ_R from Eq. 7 under two hypothetical behaviors of the learned correlation ρ. Panel 2a: $\rho(m_1)$ converges to 0, so negative correlation is not achieved and the ergodic estimator cannot outperform simple random sampling. Panel 2b: $\rho(m_1)$ decreases below zero and converges to -0.5, and the improvement ratio can fall below 1. The U–shaped improvement ratio curves illustrate the trade–off between the pilot size m_1 and the remaining budget m_2. Vertical dashed lines indicate the singularity where $m_2 = 0$; horizontal solid gray lines represent the simple random benchmark, while the dashed gray line shows the asymptotic bound

translates this behavior into the improvement ratio σ_E/σ_R from Eq. 7 which remains above 1 for all admissible m_1. The vertical dashed lines mark the singularity at which m_1 becomes so large that no budget remains for the main estimation sample ($m_2 = 0$).

Panel 2b on the right presents a more favorable situation in which $\rho(m_1)$ decreases below zero and converges to -0.5. The upper–right subplot depicts this correlation path and again indicates the zero level by a horizontal solid gray line. The lower–right subplot shows the resulting improvement ratio, which now exhibits a pronounced U–shape and allows substantial variance reduction.

The red curve corresponds to a total budget of $m = 400\,000$. In this case, although the correlation approaches the limit -0.5, the remaining m_2 is insufficient to improve upon simple random sampling; the minimum of the red curve (indicated by the marker) coincides with the horizontal solid gray line representing the simple random sampling benchmark. As the total budget m increases, the optimum (different colored markers) moves toward the asymptotic lower bound of the improvement ratio from Eq. 8, indicated by the horizontal dashed gray line.

The analysis in this section has been intentionally conceptual. It clarifies how the allocation of the computational budget between the pilot phase and the ergodic sampling phase affects the attainable variance reduction, but it does not prescribe a concrete rule for choosing m_1. The key practical questions are therefore (i) whether a sufficiently negative correlation can in fact be learned in Stage 1, and (ii) whether the resulting trade–off leaves a useful range of m_1 and m_2 in practical problems. Section 6 addresses these questions through extensive numerical experiments on eight benchmark games with widely different structural characteristics and compares the ergodic estimators with existing methods.

6 Numerical experiments

We expect the performance of the algorithm to vary depending on the specific characteristics of the underlying cooperative game. To evaluate this, we tested *OptK2Shapley* on several benchmark games commonly used in the literature (Castro et al. 2009; Maleki et al. 2014), along with three additional specific games.

6.1 Test games

Our computational study is based on eight benchmark games with markedly different distributions of marginal contributions, covering binary, highly skewed (the expected value is relatively close to its minimum and far from its maximum or vice versa), and smooth environments.

Game 1 (A non–symmetric voting game) Let $n \in \mathbb{N}$, $N = \{1, 2, \ldots, n\}$, $w = (w_1, w_2, \ldots, w_n) \in \mathbb{R}^n$ and $W \in \mathbb{R}$. For $S \subseteq N$ let

$$v(S) = \begin{cases} 1 & \text{if } \sum\limits_{i \in S} w_i > W, \\ 0 & \text{if } \sum\limits_{i \in S} w_i \leq W. \end{cases}$$

The parameters of the game are set as follows: $n = 51$, $w = (45, 41, 27, 26, 26, 25, 21, 17, 17, 14, 13, 13, 12, 12, 12, 11, \underbrace{10, \ldots, 10}_{4}, \underbrace{9, \ldots, 9}_{4}, 8, 8, \underbrace{7, \ldots, 7}_{4}, \underbrace{6, \ldots, 6}_{4}, 5, \underbrace{4, \ldots, 4}_{3}, \underbrace{3, \ldots, 3}_{7})$ and $W = \frac{\sum w_i}{2} = 269$ (this is the United States Electoral College in the '70 s).

Game 2 (A symmetric voting game) This is a special case of the non–symmetric voting game. Each weight is set to $w_i = \frac{1}{n}$ where the number of players is $n = 100$.

Game 3 (A shoes game) Let the set of players $N = \{1, 2, \ldots, 100\}$, characteristic function $v(S) = \min(S \cap N_{left}, S \cap N_{right})\ \forall S \subseteq N$ where $N_{left} = \{1, 2, \ldots, 50\}$, $N_{right} = \{51, 52, \ldots, 100\}$.

Game 4 (An airport game) Let $N = \{1, 2, \ldots, 100\}$ and $c = (\underbrace{1, \ldots, 1}_{8}, \underbrace{2, \ldots, 2}_{12}, \underbrace{3, \ldots, 3}_{6}, \underbrace{4, \ldots, 4}_{14}, \underbrace{5, \ldots, 5}_{8}, \underbrace{6, \ldots, 6}_{9}, \underbrace{7, \ldots, 7}_{13}, \underbrace{8, \ldots, 8}_{10}, \underbrace{9, \ldots, 9}_{10}, \underbrace{10, \ldots, 10}_{10})$

. The characteristic function of the game is $v(S) = \max\{c_i | i \in S\}$.

Game 5 (A minimum spanning tree game) Let G be the weighted graph on Fig. 3. The set of nodes is $V = \{0, 1, 2, \ldots, 100\}$, the edges are defined as follows:

$$E = \{(x, x+1) | x \in \{1, 2, \ldots, 99\}\} \cup \{(100, 1)\} \cup \bigcup_{i \in \{1,2,\ldots,100\}} \{(0, i)\} \subseteq V \times V.$$

All the weights on edges adjacent to node 0 are 101, and all the weights on edges that connect the circle nodes are 1. The set of players is $N = \{1, 2, \ldots, 100\}$, and the characteristic function is $v(S) =$ sum of edge weights of the minimum spanning tree of subgraph $S \cup \{0\} \subseteq G$.

Game 6 (A bankruptcy game) Let $A \in \mathbb{R}^+, n \in \mathbb{N}, \quad l_1, l_2, \ldots l_n \in \mathbb{R}^+$ and let game (N, v) be an *n* –person game where the characteristic function v is as follows: for all coalitions $S \subseteq N$ $v(S) = \max\{0, A - \sum_{j \notin S} l_j\}$. For more details on bankruptcy games, see Driessen (1988) and Thomson (2015). Let us set $n = 100$, $A = 200$ and let the vector of liabilities l_j be set to the same numbers as the c_i numbers in Game 4.

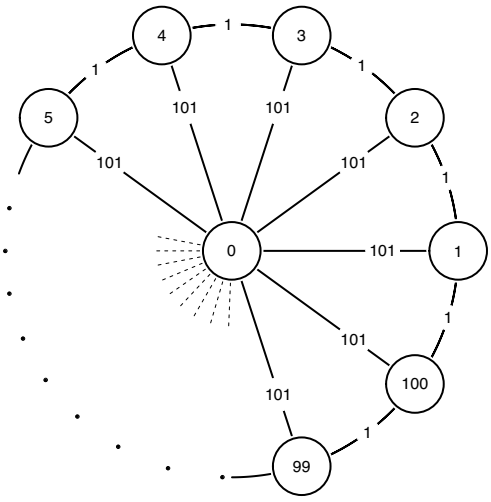

Fig. 3 Graph of minimum spanning tree game

Game 7 (A liability game) Csóka and Herings (2019) have recently introduced the liability game. Let $A \in \mathbb{R}^+, n \in \mathbb{N}, \quad l_1, l_2, \ldots l_n \in \mathbb{R}^+$ where $A < \sum l_i$, and let game (N, v) be an *n*–person game with characteristic function

$$v(S) = \begin{cases} \min\{A, \sum\limits_{j \in S \setminus \{0\}} l_j\} & \text{if } 0 \in S \\ \max\{0, A - \sum\limits_{j \notin S \cup \{0\}} l_j\} & \text{if } 0 \notin S \end{cases}.$$

According to Csóka et al. (2022), the Shapley value of liability games cannot be calculated in polynomial time (unless P=NP), so the estimation of their Shapley value is particularly interesting. In our example, we set $n = 100$, $A = 200$, and the vector of liabilities l_j are set to the same numbers as the c_i numbers in Game 4.

Game 8 (A pair game) Let $n = 100, \quad N = N_1 \cup N_2, \quad |N_1| = |N_2| = 50$ and let $G(N_1, N_2, E)$ be a bipartite graph with a complete matching (to put it simply, each player has a pair). The characteristic function of the game is $v(S) = 1/2 \cdot |(i, j) \in S \times S \text{ such that } (i, j) \in E|$, i.e. the number of pairs in the coalition.

6.2 Performance of Algorithm GreedyK2

In Sect. 5 we analysed the behaviour of algorithm *GreedyK2* in a hypothetical example, serving as a clean illustration of the underlying mechanisms. We now move beyond this stylized example and study the algorithm's behavior in practical games. While Fig. 2 depicts ρ as a smooth curve, such smoothness is not to be expected in practice. Since ρ is estimated from a finite sample, it is inherently stochastic and may display considerable variability, especially for small sample sizes.

Nevertheless, the qualitative patterns predicted by the stylized analysis–most notably the existence of a threshold for negative correlation and the resulting U-shaped trade–off in the improvement ratio–are consistent with the empirical behavior observed in Fig. 4. Deviations from the average-case curve primarily affect the location of the optimal pilot size. This variability diminishes as the pilot sample size m_1 increases, leading to more stable estimates of $\rho(m_1)$, and does not alter the qualitative conclusions of the analysis.

Figure 4 presents the results for the Symmetric voting game. The left panel displays three representative sample trajectories (shown as coloured dashed lines), together with their average over 25 runs (solid black line). In this setting, the optimal transformation *t* consists of moving the pivot player from position 51 to any position where its marginal contribution is zero. algorithm *GreedyK2* is able to identify this transformation once it encounters an ordering in which player *i* acts as the pivot player; the probability of observing such an ordering converges exponentially fast to one as the sample size increases. When this occurs, the correlation drops to its minimal value, which is close to zero, and no further improvement can be achieved.

This behavior is reflected in the lower–left subplot of Fig. 4, where the U–shaped curve of the improvement ratio attains its minimum at a value close to 1. Consequently, for this particular game, algorithm *OptK2Shapley* does not outperform the

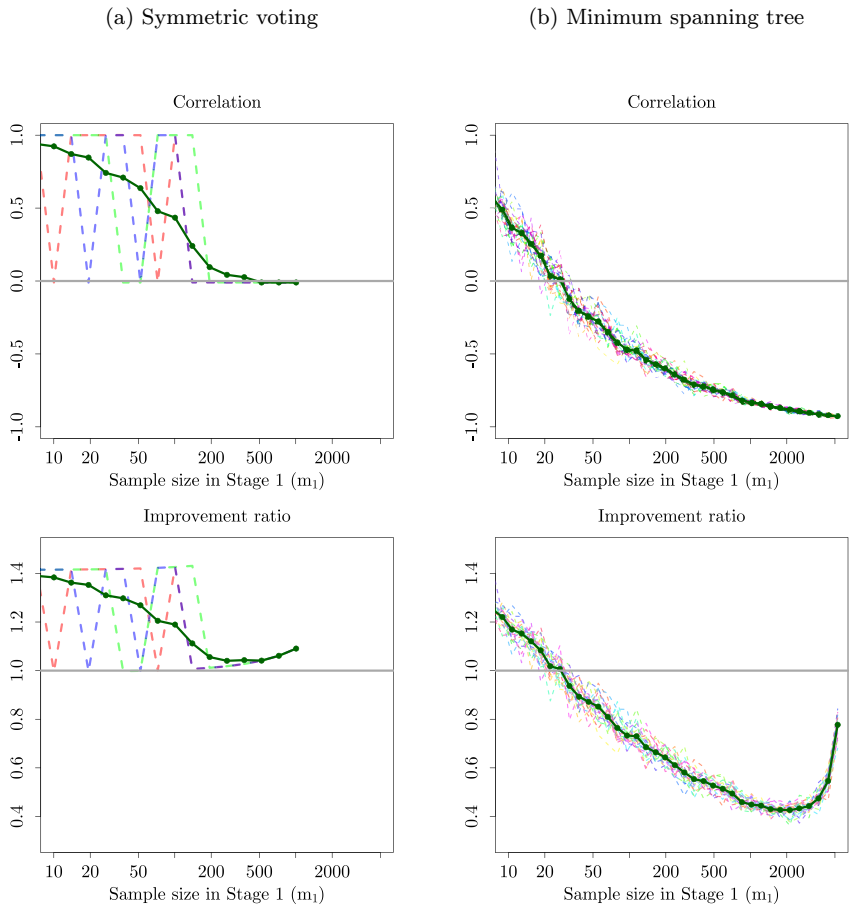

Fig. 4 Empirical analogue of Fig. 2 based on algorithm *GreedyK2*. The left panel corresponds to the Symmetric voting game. The upper plot shows the learned correlations $\rho(m_1)$ from 25 independent simulations; the solid line indicates their mean, while the dashed curves represent individual runs. Since the correlations remain non–negative, the lower plot shows that the improvement ratio σ_E/σ_R does not fall below 1, and the ergodic estimator cannot outperform simple Monte Carlo sampling. The right panel corresponds to the Minimum spanning tree game. Here, the learned correlations decrease well below zero, leading to improvement ratios below 1 and to a pronounced U–shaped trade–off between m_1 and m_2

benchmark: it produces effectively uncorrelated samples and yields asymptotically the same performance as simple random sampling.

In sharp contrast to the Symmetric voting game, the Minimum spanning tree game in Fig. 4 highlights a setting in which algorithm *GreedyK2* is highly effective. The correlation ρ remains stable across trajectories, decreases rapidly, and becomes negative already at moderate sample sizes ($m_1 \approx 50$). As a result, a significant variance reduction is achieved, which is clearly visible in the right panel of the figure.

Similar simulation experiments were conducted for the remaining test games. Overall, the empirical evidence suggests that in many games, the correlation approaches its attainable level already at moderate sizes of m_1 (around 100–200),

although in some cases substantially larger pilot samples may be required. This indicates that even a coarse but practically useful transformation can be learned from a modest number of samples, implying that the algorithm can be safely applied once the total sample size reaches a few million. Importantly, the samples expended during Stage 1 do not lead to a severe degradation of performance compared to simple random sampling, even in the worst case.

Whenever a transformation yielding significantly negative correlation exists for a given game, *Greedy2K* is usually able to identify such a transformation already at sample sizes on the order of a few hundred.

In contrast, predicting the sample size at which the correlation approaches its attainable minimum is considerably more challenging. In particular, when very strong negative correlation is achievable, learning a near–optimal transformation may require a pilot sample size that is no longer practical in real–world applications.

Out of the eight test games considered, three exhibit no improvement over simple random sampling; notably, this lack of improvement becomes apparent already at small sample sizes. These cases correspond to games in which the distribution of marginal contributions is highly asymmetric–often binary–valued–with an expected value close to zero. For the remaining five games, a substantial variance reduction is observed. However, in some of these cases, achieving the best possible transformation would require a pilot sample size that may be unrealistically large under current computational constraints. More detailed quantitative results are deferred to Sect. 6.3, where *ErgodicSampling* is also compared against alternative sampling methods.

6.3 Combined effects of variance–reduction channels

In this subsection, we examine how the effects of ergodic sampling and other variance reduction techniques operate through distinct channels and how these channels interact in the numerical experiments. Our aim is to characterize how the different variance–reduction channels combine in practice and to show that the corresponding methods can act as complementary tools.

The three procedures introduced in Sect. 2 rely on conceptually different mechanisms.

Conditional sampling reduces variance when the conditional expectations of the marginal contribution differ substantially across player positions. If conditional expectations vary widely around the unconditional expectation, then replacing empirical frequencies with known position probabilities eliminates an important source of sampling noise, yielding a large gain.

Stratified sampling can further reduce variance when the conditional standard deviations also vary with the position of the player. By allocating subsample sizes in proportion to these standard deviations, the method targets the remaining heterogeneity inside the conditional groups. Although this effect is theoretically non–zero–since the marginal contribution is constant for the first and last positions–its practical magnitude may be small when the conditional variances are nearly equal.

Ergodic sampling operates through a different channel. In its simplest form, algorithm *GreedyK2* seeks a transformation that maps positions with low conditional

expectations to positions with high ones, and vice versa[6]. The method is therefore most effective when roughly half of the conditional expectations lie below and half above the unconditional expectation, allowing negatively correlated pairs to be constructed. In such cases the residual variance–unaffected by conditioning–can be substantially reduced.

These channels are independent in a structural sense: each exploits a different property of the marginal contribution distribution, and none of these properties implies any of the others. Consequently, there is no theoretical guarantee that any single method yields a large variance reduction, and none provides a uniform improvement over simple random sampling. The magnitude of the gain can be evaluated with essentially no extra computational cost, yet it is difficult to predict ex ante which channel will dominate in a given game.

To study how these isolated mechanisms combine in practice, we conduct numerical experiments on the eight test games introduced in Sect. 6.1, using the standard Monte Carlo estimator based on simple random sampling as a baseline. We evaluate conditional random, stratified,[7] ergodic, and hybrid estimators relative to this benchmark. The hybrid methods combine the structural information exploited by conditioning or stratification with the dependence–based variance reduction of ergodic sampling.

To ensure a fair comparison, all algorithms are implemented under an identical computational budget: each method evaluates the marginal contribution of a given player exactly 10 million times. Each experiment is repeated independently 1000 times, and performance is assessed using the empirical standard deviations of the resulting estimates. For non–symmetric games, we repeat the experiments for multiple players to capture heterogeneity in marginal contribution distributions and its effect on estimator variance. All experiments were conducted in Matlab 2025b on a workstation equipped with an Intel 6–core processor (i7–9750 H) and 16 GB DDR4 RAM. The implementation details are described in the Appendix, and the full source code, together with the raw results, is available on GitHub https://github.com/kekerozberg/Estimation-of-the-Shapley-Value-by-Ergodic-Sampling.

The quantitative results in Table 2 confirm that these channels rarely appear in isolation. The eight test games exhibit different variance structures; in some games the conditioning channel explains most variability, in others the dependence channel is decisive, and in several cases both contribute.

Although the hybrid methods do not create genuine synergies–the individual channels remain conceptually independent and their effects do not combine addi-

[6] Algorithm *GreedyK2* is not limited to pairing positions based solely on conditional expectations, a task that could be accomplished in linear time, in contrast to the quadratic running time of *GreedyK2*. More generally, the conditional marginal contribution of player i may change even if its position remains fixed. For instance, in the minimum spanning tree game, modifying the relative positions of neighboring players can substantially alter the marginal contribution of player i, often with a much stronger effect than changes induced by position swaps alone.

[7] We implemented the two–stage stratified sampling algorithm of Castro et al. (2017). In the allocation stage we used the classical Neyman allocation, which assigns subsample sizes proportional to the conditional standard deviations rather than to the conditional variances. The Neyman allocation yields the optimal variance reduction.

tively–they act as true complements by preserving the advantageous features of each approach.

These results confirm the interpretation developed in Sect. 5. First, sufficiently negative correlation can be learned in several, but not all, games. Second, even when the standalone ergodic estimator is not superior, the hybrid procedures remain competitive because conditioning, stratification, and ergodic sampling exploit different sources of variability. From a practical perspective this is crucial: the proposed ergodic method should be viewed not as a replacement for existing techniques, but as an additional tool that increases robustness when the dominant source of variance is unknown ex ante.

7 Conclusion

This paper introduced an ergodic sampling approach for the estimation of the Shapley value that exploits dependence between paired marginal contributions. While classical Monte Carlo methods rely exclusively on independent sampling, the proposed estimator adapts the idea of antithetic variates to the permutation space and generates negatively correlated observations through a learned measure–preserving transformation. We proved that the resulting sequence is ergodic, unbiased, and consistent; we developed a two–stage algorithm for practical implementation.

A key insight of the analysis is the trade–off between learning quality and remaining Monte Carlo budget. The variance formula derived in Sect. 5 shows that the benefit of ergodic sampling depends on whether sufficiently negative correlation can be achieved and on how the pilot size m_1 is balanced with the main sample $2m_2$. The stylized scenarios of Fig. 2 illustrate that neither very small nor very large pilot samples are desirable, and that improvement is impossible when the attainable correlation remains non–negative.

Extensive experiments on eight benchmark games demonstrated that negative correlation can indeed be learned in several, though not all, environments. The standalone ergodic estimator performs particularly well in games with smooth and structured marginal–contribution distributions, such as the Minimum spanning tree and the Shoes games, while it offers little advantage in highly symmetric or binary settings. Most importantly, when combined with conditional and stratified sampling, the hybrid estimators were consistently close to the best available method across all games. The three variance–reduction channels operate through largely independent mechanisms, so their combination increases robustness even without creating additive synergies.

From a methodological perspective the results advocate a diagnostic approach. In practical applications it is advisable to first explore the statistical characteristics of the problem–heterogeneity of conditional means and variances, and the potential for learning negative dependence–and to select and combine variance–reduction techniques accordingly. Ergodic sampling enriches this toolbox by adding a dependence–based channel that can be activated when the structure of the game permits it, but it should be viewed as a complement rather than a replacement for existing methods.

Table 2 Improvement ratios relative to simple Monte Carlo sampling for each method and game

Game	Player	m_1	Improvement ratio				
			Conditional random	Stratified	Ergodic	Conditional ergodic	Stratified ergodic
Non–Symmetric Voting	#1	50	80.68%	55.11%	100.00%	80.54%	52.80%
	(45)		(95.00sec)	(99.40sec)	(86.22sec)	(86.22sec)	(98.41sec)
	#10	100	94.62%	61.09%	107.16%	103.29%	66.55%
	(14)		(94.88sec)	(98.98sec)	(86.23sec)	(86.23sec)	(98.63sec)
	#20	300	96.65%	67.44%	104.52%	101.17%	67.11%
	(10)		(94.29sec)	(99.14sec)	(89.14sec)	(89.14sec)	(101.42sec)
	#30	500	98.03%	62.04%	107.55%	105.33%	66.77%
	(7)		(93.29sec)	(98.10sec)	(91.59sec)	(91.59sec)	(103.51sec)
	#40	500	98.29%	62.78%	100.39%	99.87%	63.70%
	(4)		(93.78sec)	(98.43sec)	(92.27sec)	(92.27sec)	(103.71sec)
	#51	1000	99.11%	65.67%	109.09%	107.86%	66.83%
	(3)		(94.73sec)	(99.32)	(104.12sec)	(104.12sec)	(114.80sec)
Symmetric Voting	#1	500	0.00%	0.00%	100.94%	0.00%	0.00%
			(76.96sec)	(83.87sec)	(86.85sec)	(86.85sec)	(107.28sec)
Shoes	#100	1000	98.12%	92.23%	66.36%	64.91%	80.78%
			(110.56sec)	(116.92sec)	(161.59sec)	(161.59sec)	(168.60sec)
Airport	#1	300	0.00%	0.00%	103.42%	0.00%	0.00%
	(1)		(129.39sec)	(133.14sec)	(123.22sec)	(123.22sec)	(150.79sec)
	#20	300	14.06%	2.54%	104.80%	15.19%	2.55%
	(2)		(124.45sec)	(128.35sec)	(118.32sec)	(118.32sec)	(134.40sec)
	#40	150	26.58%	5.67%	103.03%	28.44%	5.92%
	(4)		(124.65sec)	(128.49sec)	(109.84sec)	(109.84sec)	(128.90sec)
	#60	150	38.27%	11.20%	105.06%	37.97%	11.44%
	(7)		(124.89sec)	(127.38sec)	(110.45sec)	(110.45sec)	(130.74sec)
	#80	150	40.14%	13.45%	101.37%	38.63%	13.98%
	(8)		(124.61sec)	(127.16sec)	(110.62sec)	(110.63sec)	(131.26sec)
	#100	100	42.15%	22.87%	98.14%	43.18%	22.32%
	(10)		(124.53sec)	(126.80sec)	(107.90sec)	(107.90sec)	(129.29sec)
Minimum Spanning Tree	#1	2000	69.23%	66.48%	45.64%	43.03%	53.62%
			(98.17sec)	(105.26sec)	(306.07sec)	(306.07sec)	(309.67sec)
Bankruptcy	#1	150	23.78%	10.32%	70.99%	23.34%	10.25%
	(1)		(126.85sec)	(132.19sec)	(112.20sec)	(112.20sec)	(138.54sec)
	#20	150	23.31%	10.53%	66.55%	24.07%	10.01%
	(2)		(127.59sec)	(132.26sec)	(112.58sec)	(112.58sec)	(137.53sec)
	#40	150	22.94%	9.85%	64.87%	22.50%	9.37%
	(4)		(127.64sec)	(132.80sec)	(113.11sec)	(113.11sec)	(137.87sec)

Table 2 (continued)

Game	Player	m_1	Improvement ratio				
			Conditional random	Stratified	Ergodic	Conditional ergodic	Stratified ergodic
	#60 (7)	150	23.31% (127.82sec)	9.75% (132.65sec)	70.08% (113.27sec)	22.30% (113.27sec)	9.55% (138.05sec)
	#80 (8)	150	22.73% (127.95sec)	9.66% (133.32sec)	65.81% (113.36sec)	22.34% (113.36sec)	9.52% (138.51sec)
	#100 (10)	80	21.49% (127.49sec)	9.44% (132.51sec)	69.50% (109.33sec)	20.46% (109.33sec)	8.64% (134.58sec)
Liability	#0	2000	13.79% (138.23sec)	11.66% (143.71sec)	40.77% (339.16sec)	14.81% (339.16sec)	13.50% (344.63sec)
Pair	#100	2800	81.96% (88.21sec)	78.75% (95.46sec)	50.57% (430.09sec)	49.07% (430.09sec)	56.75% (429.61sec)

Values report $\sigma_{\text{method}}/\sigma_R$; numbers below 1 indicate variance reduction, while values above 1 correspond to a deterioration. All estimators are evaluated under an identical computational budget of 10 million marginal–contribution evaluations, including the pilot phase of the ergodic methods. Each experiment was repeated independently 1000 times. For non–symmetric games the player's weight is reported in parentheses. Running times of an experiment are shown in parentheses below the improvement ratios

Note: Reported running times are illustrative only. Although the computational budget is identical across methods, wall–clock times reflect implementation details rather than algorithmic complexity. The Ergodic and Conditional Ergodic estimators were computed simultaneously and therefore have identical reported times

Several directions remain for future research. First, adaptive strategies for choosing the pilot size m_1 based on online estimates of the correlation could further improve efficiency. Second, extensions to transformations of higher order ($K > 2$) may capture more complex dependence patterns. Finally, the idea of learning measure–preserving transformations may be useful beyond Shapley value estimation, for example in other permutation–based Monte Carlo problems.

A Proofs

Proof of Theorem 1 The measure of any cylinder set $C = (C_1, C_2, \dots) \in \mathcal{A}$ is $\mathbb{P}(C) = \left(\frac{1}{|\Omega_1|}\right)^{|\{j|C_j \neq \Omega_1\}|}$. As t is surjective, $C_j = \Omega_1 \Leftrightarrow t(C_j) = \Omega_1$, therefore

$\mathbb{P}(C) = \mathbb{P}(T(C))$. This means that T is measure-preserving (it is enough to check this on the cylinders), which is the key to the proof. At first, we have to show that the variables are identically distributed

$$\begin{aligned}\mathbb{P}(Y_{k,l} \in B) &= \mathbb{P}\left(\{\omega \in \Omega | \omega \in Y_{k,l}^{-1}(B)\}\right) = \mathbb{P}\left(\{\omega \in \Omega | T^{k-1}(\omega) \in Y_{1,l}^{-1}(B)\}\right) = \\ &= \mathbb{P}\left(\{T^{k-1}(\omega) | T^{k-1}(\omega) \in Y_{1,l}^{-1}(B)\}\right) = \mathbb{P}\left(\{\omega \in \Omega | \omega \in Y_{1,l}^{-1}(B)\}\right) = \\ &= \mathbb{P}(Y_{1,l} \in B) = \mathbb{P}(X_l \in B) = \mathbb{P}(X_1 \in B)\end{aligned}$$

for all $B \subseteq \mathbb{R}$ Borel-set. The third equation uses that T is measure-preserving, the other equations are trivial. Now let $m \in \mathbb{N}$ be an integer such that $K|m$[8] and let $m = KL$.

$$\begin{aligned}&\frac{\sum Y_i}{m} = \frac{Y_1 + Y_2 + \cdots Y_m}{m} \\ &\frac{(Y_{1,1} + Y_{2,1} + \cdots + Y_{K,1}) + (Y_{1,2} + \cdots + Y_{K,2}) + \cdots + (Y_{1,L} + \cdots + Y_{K,L})}{KL} \\ &= \frac{\frac{(Y_{1,1}+Y_{1,2}+\cdots+Y_{1,L})}{L} + \frac{(Y_{2,1}+Y_{2,2}+\cdots+Y_{2,L})}{L} + \frac{(Y_{K,1}+Y_{K,2}+\cdots+Y_{K,L})}{L}}{K} = \frac{Z_1 + Z_2 + \cdots + Z_K}{K}\end{aligned}$$

It is enough to show that for $\forall\, 1 \leq k \leq K$ variables $Y_{k,1}, Y_{k,2}, \dots, Y_{k,L}$ are independent. In that case $\mathbb{P}(Z_k \to \mu) = 1$, therefore $\mathbb{P}\left(\bigcap_{k=1}^{K} \{\omega \in \Omega | Z_k(\omega) \to \mu\}\right) = 1$

. Obviously, the mean of finitely many convergent series with the same limit is convergent to the same number, so $\frac{\sum Y_i}{m} \to \mu$. □

[8] Notation $K|m$ stands for K is a divisor of m

Remark The independence of variables $Y_{k,1}, Y_{k,2}, \ldots, Y_{k,L}$ implies that $\sqrt{L}(Z_j - \mu)$ is asymptotically normally distributed for all j by the central limit theorem. The same argument does not hold for the mean of the $Z_j s$ because they are not independent but negatively correlated. However, excluding degenerate cases, for example, when the variance of the sample mean is completely eliminated (it is possible), if there is any limit distribution for the sample mean, it can be nothing but normal.

Variables $Y_{k,1}, Y_{k,2}, \ldots, Y_{k,L}$ are by definition independent for $k = 1$. For $k \geq 2$, this can be proved as follows:

$$\mathbb{P}\left(\bigcap_{j=1}^{L}\{Y_{k,j} \in B_j\}\right) = \mathbb{P}\left(\left\{\omega \in \Omega | \omega \in Y_{k,1}^{-1}(B_1), \omega \in Y_{k,2}^{-1}(B_2), \ldots, \omega \in Y_{k,L}^{-1}(B_L)\right\}\right)$$
$$= \mathbb{P}\left(\left\{\omega \in \Omega | T^{k-1}(\omega) \in Y_{1,1}^{-1}(B_1), T^{k-1}(\omega) \in Y_{1,2}^{-1}(B_2), \ldots, T^{k-1}(\omega) \in Y_{k,L}^{-1}(B_L)\right\}\right)$$
$$= \mathbb{P}\left(\left\{T^{k-1}(\omega) | T^{k-1}(\omega) \in Y_{1,1}^{-1}(B_1), T^{k-1}(\omega) \in Y_{1,2}^{-1}(B_2), \ldots, T^{k-1}(\omega) \in Y_{k,L}^{-1}(B_L)\right\}\right)$$
$$= \mathbb{P}\left(\left\{\omega \mid \omega \in Y_{1,1}^{-1}(B_1), \omega \in Y_{1,2}^{-1}(B_2), \ldots, \omega \in Y_{k,L}^{-1}(B_L)\right\}\right) =$$
$$= \prod_{j=1}^{L} \mathbb{P}\left(\{\omega \in \Omega \,|\, \omega \in Y_{1,j}^{-1}(B_j)\}\right) = \prod_{j=1}^{L} \mathbb{P}\left(Y_{1,j} \in B_j\right) = \prod_{j=1}^{L} \mathbb{P}\left(Y_{k,j} \in B_j\right).$$

For all $B_1, B_2, \ldots B_L \subseteq \mathbb{R}$ Borel-sets. Here, the third line comes from the fact that T is measure-preserving, and the last equation holds because we have shown that all the variables are identically distributed.

Now assume that $K \not| m$, and let $s = s(m) = \max\{j \leq m \mid K|j\}$.

$$\frac{Y_1 + Y_2 + \ldots Y_m}{m} = \frac{Y_1 + Y_2 + \cdots Y_s}{m} + \frac{Y_{s+1} + \cdots + Y_m}{m} =$$
$$= \underbrace{\frac{Y_1 + Y_2 + \cdots Y_s}{s}}_{\downarrow \mu} \cdot \underbrace{\frac{s}{m}}_{\downarrow 1} + \underbrace{\frac{Y_{s+1} + \cdots + Y_m}{m}}_{\downarrow 0}$$

As K is a constant and $m - s \leq K$ if $m \to \infty$, then so does s, so the first term goes to μ and $\frac{s}{m} \to 1$. The last term on the right side is bounded by $\frac{K \cdot \max|X|}{m} \to 0$. □

Proof of Theorem 3 Consider the elements of Ω_1, sorted according to the X values, and define t such that this order is reversed (i.e., the first element is assigned to the last and vice versa, etc.). Without loss of generality, we can assume that X has a mean of 0; in this case, the correlation and sumproduct have the same sign. It therefore suffices to show the following:

Let $a_1 \leq a_2 \leq \cdots \leq a_L$ be a monotone sequence of real number such that

$$\sum a_i = 0.$$

Then

$$\sum a_i a_{L+1-i} \leq 0.$$

And this is easy to prove as follows.

$$\begin{aligned}
0 &\geq \sum_i \sum_j \underbrace{(a_i - a_j)(a_{L+1-i} - a_{L+1-j})}_{\text{These terms always have opposite signes.}} \\
&= \sum_i \sum_j a_i a_{L+1-i} - \sum_i \sum_j a_i a_{L+1-j} - \sum_i \sum_j a_j a_{L+1-i} + \sum_i \sum_j a_j a_{L+1-j} \\
&= 2L \sum_i a_i a_{L+1-i} - \sum_i a_i \underbrace{\sum_j a_{L+1-j}}_{0} - \sum_i a_{L+1-i} \underbrace{\sum_j a_j}_{0}.
\end{aligned}$$

□

B Algorithms

B.1 Ergodic sampling

```
Parameters:
    m2: Integer   -- half of the sample size
    t: Function   -- transformation
    M: Function   -- marginal contribution
    n: Integer    -- number of players
    i: Integer    -- player encoded as an integer

marginal_contributions <- Array[1:2*m2]
For j = 2 To 2*m2 Step 2:
    o1 <- Shuffle(n) # a random permutation of set [1,2,...,n]
    o2 <- t(o1)
    marginal_contribution1 <- M(o1, i)
    marginal_contribution2 <- M(o2, i)
    marginal_contributions[j-1] <- marginal_contribution1
    marginal_contributions[j] <- marginal_contribution2
EndFor

Return Mean(marginal_contributions)
```

Algorithm 2: GreedyK2

```
Parameters:
    m1: Integer   -- sample size
    M: Function   -- marginal contribution
    n: Integer    -- number of players
    i: Integer    -- player encoded as an integer

orders <- Matrix(0, rows = 1:m1, cols = 1:n)
For j = 1 To m1:
    orders.Row[j] <- Shuffle(n)
EndFor

covars = Matrix(0, rows = 1:n, cols = 1:n)
mc_base <- Array[1:m1]

For j = 1 To m1:
    o <- shuffle(n)
    mc_base[j] <- M(o, i)
EndFor

mc_swap <- Array[1:m1]

For left = 1 To n-1:
    For right = left To n:
        swapped_orders <- orders
        swapped_orders.Swap_columns(left, right)
        For k = 1 To m1:
            player_pos <- orders.Row[k].Index(i)
            If left <= player_pos And player_pos <= right:
                mc_swap[k] <- M(swapped_orders.Row[k], i)
            Else:
                mc_swap[k] <- mc_base[k]  # it does not change
            EndIf
        EndFor
        covars[i, j] <- covars[j, i] <- Cov(mc_base, mc_swap) # covariance
    EndFor
EndFor

transformation <- Array[1:n]

While covars.Size > 0:
    r, c = Which_min(covars) # row and column index
    transformation[r] <- c
    transformation[c] <- r
    covars.Delete_row(r)
    covars.Delete_col(c)
EndWhile

Return transformation
```

B.2 Conditional ergodic sampling

1. *Stage 1*Choose a suitable sample size $m_1 \in \mathbb{N}$.
2. Generate m_1 i.i.d. player orders $o_1, o_2, \ldots o_{m_1}$, with replacement. Let $\mathcal{O}$ denote this sample of orders.
3. For each $a \leq b \in \{1, 2, \ldots, n\}$, define $\mathcal{O}_{a,b} = \{\langle a, b\rangle \circ o_j \,|\, o_j \in \mathcal{O}\}$.
4. For each such pair, compute the empirical covariance $c_{a,b}$ between the samples $\{M_o(i) \,|\, o \in \mathcal{O}\}$ and $\{M_o(i) \,|\, o \in \mathcal{O}_{a,b}\}$

5. Sort all $c_{a,b}$ in ascending order: $c_{a_1,b_1} \leq c_{a_1,b_2} \leq \ldots \leq c_{a_n,b_n}$. The length of this sequence is $\binom{n}{2}$.
6. Initialize an empty list E_{op}. Iteratively select the lowest covariance pair (a_k, b_k) that does not overlap with already chosen indices, and add it to E_{op}.
7. Define $\pi = \prod_{(i,j)\in E_{op}} \langle i, j\rangle$, and let $t_\pi(o) = \pi \circ o$ for all $o \in O_N$.

Stage 2

1. Generate an initial random player order $o_1 \sim \text{Uniform}(O_N)$.
2. Transform it into a new order $o_2 = t(o_1) \in O_N$
3. Compute and store the marginal contributions of player i for both orders $Y_1 = M_{o_1}(i)$ and $Y_2 = M_{o_2}(i)$.
4. Repeat the process m_2 times.
5. Compute the conditional mean of the entire sample.

B.3 Stratified ergodic sampling

Stage 1

1. Choose a suitable sample size $m_1 \in \mathbb{N}$.
2. Generate m_1 i.i.d. player orders $o_1, o_2, \ldots o_{m_1}$, with replacement. Let $\mathcal{O}$ denote this sample of orders.
3. For each $a \leq b \in \{1, 2, \ldots, n\}$, define $\mathcal{O}_{a,b} = \{\langle a, b\rangle \circ o_j \mid o_j \in \mathcal{O}\}$.
4. For each such pair, compute the empirical covariance $c_{a,b}$ between the samples $\{M_o(i) \mid o \in \mathcal{O}\}$ and $\{M_o(i) \mid o \in \mathcal{O}_{a,b}\}$
5. Sort all $c_{a,b}$ in ascending order: $c_{a_1,b_1} \leq c_{a_1,b_2} \leq \cdots \leq c_{a_n,b_n}$. The length of this sequence is $\binom{n}{2}$.
6. Initialize an empty list E_{op}. Iteratively select the lowest covariance pair (a_k, b_k) that does not overlap with already chosen indices, and add it to E_{op}.
7. Define $\pi = \prod_{(i,j)\in E_{op}} \langle i, j\rangle$, and let $t_\pi(o) = \pi \circ o$ for all $o \in O_N$.

Stage 2

1. Generate new random player orders independently for each position *l* (*n* times)
2. Replace the player *i* with the player in position *l* for each *l*. It guarantees that we have the same number of orders where the player *i* is in the position *l*.
3. Compute the player *i*'s marginal contributions in each orders.
4. Calculate the standard deviations of the marginal contributions of the player *i* by each position *l*.
5. The ratio of the standard deviations determines how many ergodic samples are needed to generate for each position in Stage 3.

Stage 3

1. Generate an initial random player order $o_1 \sim \text{Uniform}(O_N)$.
2. Replace the player i with the player in position l for each l.
3. Transform it into a new order $o_2 = t(o_1) \in O_N$
4. Compute and store the marginal contributions of player i for both orders $Y_1 = M_{o_1}(i)$ and $Y_2 = M_{o_2}(i)$.
5. Repeat the process for each position l as many times as determined by Stage 3.
6. Compute the conditional mean of the entire sample.

Funding Open access funding provided by Corvinus University of Budapest.

Declarations

Conflict of interest The authors have no conflict of interest to declare that are relevant to the content of this article.

References

Mann I, Shapley LS (1960) Values of large games, iv: evaluating the electoral college by Monte Carlo techniques. Technical Report RM-2651, RAND Corporation, Santa Monica, CA. https://www.rand.org/pubs/research_memoranda/RM2651.html

Shapley LS (1953) A value for n-person games. Contrib Theory Games 2(28):307–317

Shapley LS, Shubik M (1969) On market games. J Econ Theory 1(1):9–25. https://doi.org/10.1016/0022-0531(69)90008-8

Amihud Y, Barnea A (1974) Portfolio selection for managerial control. Omega 2(6):775–783. https://doi.org/10.1016/0305-0483(74)90116-9

Csóka P, Herings PJ-J, Kóczy LÁ (2009) Stable allocations of risk. Games Econom Behav 67(1):266–276. https://doi.org/10.1016/j.geb.2008.11.001

Colini-Baldeschi R, Scarsini M, Vaccari S (2018) Variance allocation and Shapley value. Methodol Comput Appl Probab 20:919–933. https://doi.org/10.1007/s11009-016-9540-5

Dehez P, Ferey S (2013) How to share joint liability: a cooperative game approach. Math Soc Sci 66(1):44–50. https://doi.org/10.1016/j.mathsocsci.2013.02.003

Ferey S, Dehez P (2016) Multiple causation, apportionment, and the Shapley value. J Leg Stud 45(1):143–171. https://doi.org/10.1086/685940

Pintér M (2011) Regression games. Ann Oper Res 186:263–274. https://doi.org/10.1007/s10479-011-0897-0

Lundberg SM, Lee S-I (2017) A unified approach to interpreting model predictions. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds.) Advances in neural information processing systems, vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

Ancona M, Oztireli C, Gross M (2019) Explaining deep neural networks with a polynomial time algorithm for Shapley value approximation. In: Chaudhuri K, Salakhutdinov R (eds.) Proceedings of the 36th international conference on machine learning. Proceedings of machine learning research, vol 97. PMLR, pp 272–281. https://proceedings.mlr.press/v97/ancona19a.html

Driessen TS (1988) Cooperative games, solutions and applications, vol 3. Springer, Dordrecht. https://doi.org/10.1007/978-94-015-7787-8

Ieong S, Shoham Y (2005) Marginal contribution nets: a compact representation scheme for coalitional games. In: Proceedings of the 6th ACM conference on electronic commerce, pp 193–202. https://doi.org/10.1145/1064009.1064030

Megiddo N (1978) Computational complexity of the game theory approach to cost allocation for a tree. Math Oper Res 3(3):189–196. https://doi.org/10.1287/moor.3.3.189

Granot D, Kuipers J, Chopra S (2002) Cost allocation for a tree network with heterogeneous customers. Math Oper Res 27(4):647–661. https://doi.org/10.1287/moor.27.4.647.307

Castro J, Gómez D, Tejada J (2008) A polynomial rule for the problem of sharing delay costs in PERT networks. Comput Oper Res 35(7):2376–2387. https://doi.org/10.1016/j.cor.2006.11.003

Deng X, Papadimitriou CH (1994) On the complexity of cooperative solution concepts. Math Oper Res 19(2):257–266. https://doi.org/10.1287/moor.19.2.257

Faigle U, Kern W (1992) The Shapley value for cooperative games under precedence constraints. Internat J Game Theory 21(3):249–266. https://doi.org/10.1007/BF01258278

Wooldridge M, Dunne PE (2006) On the computational complexity of coalitional resource games. Artif Intell 170(10):835–871. https://doi.org/10.1016/j.artint.2006.03.003

Bachrach Y, Rosenschein JS (2008) Coalitional skill games. In: Proceedings of the 7th international joint conference on autonomous agents and multiagent systems-volume 2, pp 1023–1030. https://www.ifaamas.org/Proceedings/aamas08/proceedings/pdf/paper/AAMAS08_0035.pdf

Csóka P, Illés F, Solymosi T (2022) On the Shapley value of liability games. Eur J Oper Res 300(1):378–386. https://doi.org/10.1016/j.ejor.2021.10.012

Castro J, Gómez D, Tejada J (2009) Polynomial calculation of the Shapley value based on sampling. Comput Oper Res 36(5):1726–1730. https://doi.org/10.1016/j.cor.2008.04.004

Benati S, López-Blázquez F, Puerto J (2019) A stochastic approach to approximate values in cooperative games. Eur J Oper Res 279(1):93–106. https://doi.org/10.1016/j.ejor.2019.05.027

Armijos-Toro L, Alonso-Meijide J, Mosquera M, Saavedra-Nieves A (2024) A sampling approach for the approximation of the Deegan–Packel index. Expert Syst Appl 256:124876. https://doi.org/10.1016/j.eswa.2024.124876

Touati S, Radjef MS, Sais L (2021) A Bayesian Monte Carlo method for computing the Shapley value: application to weighted voting and bin packing games. Comput Oper Res 125:105094. https://doi.org/10.1016/j.cor.2020.105094

Maleki S, Tran-Thanh L, Hines G, Rahwan T, Rogers A (2014) Bounding the estimation error of sampling-based Shapley value approximation. arxiv:1306.4265

Castro J, Gómez D, Molina E, Tejada J (2017) Improving polynomial estimation of the Shapley value by stratified random sampling with optimum allocation. Comput Oper Res 82:180–188. https://doi.org/10.1016/j.cor.2017.01.019

Okhrati R, Lipani A (2021) A multilinear sampling algorithm to estimate shapley values. In: 2020 25th international conference on pattern recognition (ICPR). IEEE, pp 7992–7999. https://doi.org/10.1109/ICPR48806.2021.9412511

Staudacher J, Pollmann T (2023) Assessing antithetic sampling for approximating Shapley, Banzhaf, and Owen values. Appl Math 3(4):957–988. https://doi.org/10.3390/appliedmath3040049

Mitchell R, Cooper J, Frank E, Holmes G (2022) Sampling permutations for Shapley value estimation. J Mach Learn Res 23(43):1–46

Thomson W (2015) Axiomatic and game-theoretic analysis of bankruptcy and taxation problems: an update. Math Soc Sci 74:41–59. https://doi.org/10.1016/S0165-4896(02)00070-7

Csóka P, Herings PJ-J (2019) Liability games. Games Econ Behav 116:260–268. https://doi.org/10.1016/j.geb.2019.05.007