BETTER THAN JUST AVERAGE: COMPUTING AND USING SHAPLEY VARIANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Shapley values summarize a player's marginal contributions across coalitions by their mean, potentially hiding large, inconsistent effects behind a constant attribution. Shapley variance complements this mean by measuring the squared variation of marginal contributions around it. Despite this natural interpretation, Shapley variance remains rarely used in explainable AI, in part because tools for computing it are limited. We develop exact and efficient algorithms for games represented by sparse polynomials, tree ensembles, and Gaussian processes with product-Hamming kernels. For general games, we build on recent surrogate-based approaches to Shapley value estimation, using our exact algorithms on learned surrogates to efficiently estimate Shapley variance. On local explanation games, for example, our polynomial estimator achieves 16-fold lower median squared error than the prior Monte Carlo estimator. Beyond enabling computation, our work demonstrates how Shapley variance also provides a new lens on Shapley value estimation itself. We show that complementary paired sampling reduces the error of Monte Carlo estimation exactly when the game's odd component has less total Shapley variance than its even component, and derive a corresponding residual decomposition for KernelSHAP and LeverageSHAP. Together, these results make Shapley variance practical to compute and reveal how variation in marginal contributions governs the behavior of widely used Shapley value estimators. We hope our work enables explanations that characterize not only the mean marginal contribution but also more accurately summarizes their distribution.

1 INTRODUCTION

Predictive algorithms inform consequential decisions in finance, healthcare, and the justice system (Bartlett et al., 2022; Obermeyer et al., 2019; Imai et al., 2023). Understanding these systems requires methods for identifying how their inputs, training data, and internal components contribute to their predictions and performance. *Shapley values* provide a widely used framework for quantifying these contributions (Shapley, 1953; Lundberg & Lee, 2017; Rozemberczki et al., 2022). They attribute a system's output or performance to individual contributors by averaging their marginal contributions across subsets. This framework supports applications ranging from explaining individual predictions (Lundberg & Lee, 2017) and valuing training data (Ghorbani & Zou, 2019) to identifying influential neurons (Ghorbani & Zou, 2020) and attributing outcomes to causal variables (Janzing et al., 2020).

In this framework, d is the number of players, e.g., features, and $[d] = \{1, \dots, d\}$. Consider the game $f : 2^{[d]} \to \mathbb{R}$ assigning a real value to each coalition $S \subseteq [d]$ of players, e.g., the prediction a model makes when features in S are revealed. For a coalition $S \subseteq [d] \setminus \{i\}$, the discrete derivative measures player i's marginal contribution:

$$\Delta_i f(S) = f(S \cup \{i\}) - f(S). \tag{1}$$

In words, the marginal contribution describes the effect of a player when added to a coalition of players S.

Since there are 2^d coalitions, it is generally infeasible to inspect every marginal contribution. The Shapley value summarizes the distribution by its expectation under the *Shapley distribution*:

$$\phi_i(f) = \sum_{S \subseteq [d] \setminus \{i\}} \Delta_i f(S) \cdot \frac{1}{d\binom{d-1}{|S|}} = \mathbb{E}_S[\Delta_i f(S)]. \tag{2}$$

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



Figure 1: Shapley variance reveals context dependence hidden by similar mean attributions: bedrooms and patch A vary predictions more than rooms and patch B. For this seven-room, three-bedroom house, including bedrooms raises the predicted price by $1.5k alone but lowers it by $2.9k when floor area is already included. Patch A can raise or lower Egyptian-cat probability depending on the other visible patches. **Higher variance flags interactions that a constant attribution misses**; transparency indicates when the mean is a poor approximation to the player's effect.

The Shapley value is the best constant approximation to marginal contributions under squared error:

$$\phi_i(f) = \arg\min_{c \in \mathbb{R}} \mathbb{E}_S[(\Delta_i f(S) - c)^2]. \tag{3}$$

Although natural and useful, the Shapley value can hide how a player's effect depends on the other players present. A player with large, opposing effects can have a Shapley value near zero, just like a player with uniformly small effects (Kumar et al., 2021; Franceschi et al., 2024; Bilodeau et al., 2024; Huang & Marques-Silva, 2024; Fryer et al., 2021). More generally, **the mean alone gives no indication of how much the marginal contribution varies across coalitions**: two players can have the same Shapley value and very different distributions of effects (Kargin, 2005).

In order to address these limitations, we turn to the *Shapley variance* (Kargin, 2005):

$$\sigma_i^2(f) := \mathrm{Var}_S(\Delta_i f(S)) = \mathbb{E}_S\left[(\Delta_i f(S) - \phi_i(f))^2\right]. \tag{4}$$

As noted by Ortmann (2016, Section 2.4), the **Shapley variance measures the mean squared error of using the Shapley value as a constant approximation** to marginal contributions. Figure 1 adds this information to familiar Shapley value visualizations: transparency indicates high variance relative to the squared Shapley value.

Shapley variance was introduced to quantify uncertainty in a player's marginal contribution (Kargin, 2005). It has been studied for quadratic (Ortmann, 2016) and voting (Fatima et al., 2006) games, and computed for feature-importance (Heese et al., 2023) and quantum-circuit (Heese et al., 2025) explanations. Related work studies marginal-contribution distributions and their moments (Heese et al., 2023), including extensions to models that output probability distributions (Franceschi et al., 2024). However, to the best of our knowledge, the Shapley variance is rarely used in explainable AI.

A **practical barrier to using the Shapley variance is computing it**. For unrestricted, unbounded games, even a fixed additive-error or constant-factor approximation to Shapley variance with constant

success probability requires $\Omega(2^d)$ value queries in the worst case (Appendix G.1). Beyond game-specific and exhaustive methods, we are only aware of **a single existing method for estimating Shapley variance**: the natural permutation Monte Carlo estimator (MC) (Goldwasser & Hooker, 2024, Section 4.1). In contrast, there is a rich body of work on estimating Shapley values (Fumagalli et al., 2026; Musco & Witter, 2025; Zhou et al., 2026; Witter et al., 2025; Mitchell et al., 2022).

1.1 Our Contributions

We provide efficient algorithms for computing and estimating the Shapley variance, and demonstrate how analyzing the Shapley variance can give insights into Shapley value estimation.

We introduce **efficient algorithms to exactly compute the Shapley variance structured games** like sparse Fourier and Möbius representations, tree ensembles like XGBoost (Chen & Guestrin, 2016) and LightGBM (Ke et al., 2017), and product-Hamming Gaussian process representations.

We also develop **estimators for general, unstructured games**. In particular, we borrow a natural idea from Shapley value estimation: fit a structured surrogate $\hat{f}$ to f on samples and return the Shapley variance of the surrogate with a correction term (Witter et al., 2025). Our algorithms outperform the prior Monte Carlo (MC) estimator, sometimes by orders of magnitude. On local explanation games, for example, the polynomial estimator's median error is 16-fold lower than that of MC (Table 1).

Our relationship to the Shapley value estimation literature is synergistic: As noted by prior work (Wu et al., 2023), Shapley variance precisely controls the expected squared error of the standard Shapley value Monte Carlo estimator. We leverage this insight, together with a decomposition of variance in terms of the odd and even components of the game, to answer **exactly when the popular *paired sampling* heuristic improves Shapley value estimator performance**. We also generalize the characterization to weighted least-squares Shapley value estimators, the formulation underlying KernelSHAP and LeverageSHAP (Lundberg & Lee, 2017; Musco & Witter, 2025).

By enabling more efficient computation and demonstrating insights from Shapley variance, we hope our work supports further exploration of the marginal contribution distribution beyond its expectation.

1.2 Related Work

It is well known that Shapley values can poorly approximate a game. Shapley residual norms provide a scalar summary of how accurately the Shapley value approximates the game (Kumar et al., 2021). Interestingly, we find two games with equal raw and normalized residual norms for one player but a twofold variance difference (Appendix A.4). In addition, the proposed method for computing the residual norm evaluates the full coalition game and solves a least-squares problem on its hypercube, with exponential cost in d (Kumar et al., 2021, Sections 3, 4.1, and 7).

In addition, the Shapley values only summarize the contribution of a player in isolation. *Shapley interactions* attribute effects to sets of players, using averaged discrete derivatives (Grabisch & Roubens, 1999; Sundararajan et al., 2020) or weighted polynomial regression (Tsai et al., 2023). They provide a richer description than individual Shapley values, but there are $\binom{d}{k}$ interactions of size k. We note that variance is an orthogonal extension and can be naturally applied to Shapley interactions, i.e., the *Shapley interaction variance* describes how well the average marginal contribution of a coalition approximates its distribution.

While principled, the Shapley distribution is not alone. *Probabilistic values* average marginal contributions under alternative coalition distributions (Weber, 1988). Examples include Banzhaf values, which weight all coalitions excluding the player equally (Banzhaf, 1965), weighted Banzhaf values (Li & Yu, 2023), and Beta Shapley (Kwon & Zou, 2022). Choosing coalition weights requires an additional criterion; for example, Beta Shapley's parameters can emphasize sizes with a better signal-to-noise ratio (Kwon & Zou, 2022). As for Shapley interactions, we note that variance is an orthogonal extension and can naturally be applied to distributions beyond the Shapley distribution.

2 Preliminaries

We will use two polynomial bases throughout our work:

Definition 2.1 (Fourier Expansion)**.** For $S, T \subseteq [d]$, let $z_j(S) = 2\mathbf{1}\{j \in S\} - 1$ and $\chi_T(S) = \prod_{j \in T} z_j(S)$. The Walsh expansion (O'Donnell, 2014) of $f : 2^{[d]} \to \mathbb{R}$ is

$$f(S) = \sum_{T \subseteq [d]} F(T)\chi_T(S), \qquad F(T) = 2^{-d} \sum_{S \subseteq [d]} f(S)\chi_T(S).$$

Definition 2.2 (Möbius Expansion)**.** The Möbius monomial for $T \subseteq [d]$ is $\prod_{j \in T} \mathbf{1}[j \in S] = \mathbf{1}[T \subseteq S]$. Every game $f : 2^{[d]} \to \mathbb{R}$ has the expansion

$$f(S) = \sum_{T \subseteq S} m(T), \qquad m(T) = \sum_{U \subseteq T} (-1)^{|T|-|U|} f(U).$$

The sums include the empty set, with $m(\emptyset) = f(\emptyset)$.

3 Computing Shapley Variance

The practical barrier to using the Shapley variance is computing it. Even randomized approximation of unrestricted games requires exponentially many queries for fixed additive-error or constant-factor guarantees (Appendix G.1). In this section, we give exact computations for several structured representations.

3.1 Polynomial Representations

We consider two polynomial representations: The Fourier terms are orthogonal under the uniform coalition distribution, while the Möbius terms naturally represent the hierarchies common in games from machine learning applications (Butler et al., 2025). We show that the Shapley variance admits a quadratic form in both.

Lemma 3.1 (Fourier Representation)**.** *The Shapley variance is a quadratic form over Fourier coefficients as*

$$\sigma_i^2(f) = 4 \sum_{T,T' \ni i:\, |T| \equiv |T'| \pmod 2} F(T)F(T') \left[\frac{1}{|T \triangle T'| + 1} - \frac{\mathbf{1}[\,|T| \text{ odd}\,]}{|T|\,|T'|} \right]. \tag{5}$$

Lemma 3.2 (Möbius Representation)**.** *The Shapley variance is a quadratic form over Möbius coefficients as*

$$\sigma_i^2(f) = \sum_{T,T' \ni i} m(T)m(T') \left[\frac{1}{|T \cup T'|} - \frac{1}{|T|\,|T'|} \right]. \tag{6}$$

where the sum runs over ordered pairs $T, T' \subseteq [d]$ containing i.

For symmetric quadratic games, this specializes to the variance formula of Ortmann (2016, Lemma 2).

While the Möbius representation includes every pair of coefficients involving the player, the Fourier representation only includes pairs whose index sets have the same parity.

Consequently, computing Shapley variance from a supplied sparse representation takes quadratic time in its number of terms.

Corollary 3.3. *Given the support and coefficients of an r-sparse Fourier or Möbius representation of f, its Shapley variance vector can be computed in $O(d \cdot r^2)$ time.*

Appendix A.1 proves these representations. We will apply Corollary 3.3 to compute the Shapley variance of tree-ensembles in the next subsection and use both equations to build efficient surrogate-based estimators in Section 4.

3.2 Tree Representation

We present an algorithm to compute the exact variance of a tree-ensemble game: expand the leaves into Fourier coefficients, then evaluate the variance as a quadratic form.

Our algorithm extends the tree-specific computation of Shapley values in TreeSHAP (Lundberg et al., 2020) to variance, and directly applies to powerful and popular methods like LIGHTGBM and XGBOOST, once the coalition game is expressed in this form.

A tree leaf fixes coalition signs $z_j = 2\mathbf{1}[j \in S] - 1 = s_j$ on a set P of features. Expanding these leaf indicators gives the ensemble's exact Fourier coefficients (Gorji et al., 2025, Section 3.2) as described in Algorithm 1.

Algorithm 1 Exact Variance from Tree Leaves

Require: A tree-ensemble game f on coalition indicators.
1: Initialize a Fourier coefficient map $F \leftarrow 0$.
2: **for** each reachable leaf with weighted value v, constrained features P, and signs s_j **do**
3: **for** each $T \subseteq P$ **do**
4: $F(T) \leftarrow F(T) + v\, 2^{-|P|} \prod_{j \in T} s_j$.
5: **return** $\boldsymbol{\sigma}^2(f) = (\sigma_i^2(f))_{i=1}^d$ from Lemma 3.1, using the merged map F.

Lemma 3.4 (Exact Tree Computation)**.** *Suppose the ensemble has L leaves, at most h distinct constrained features per path, and r nonzero merged Fourier coefficients. Then $r \leq L2^h$, and Algorithm 1 computes the exact Shapley variance vector in $O(dL2^h + dr^2)$ time.*

From Tree-Ensemble Models to Tree-Ensemble Games. For a fitted tree ensemble model, we define a coalition game by keeping the features in the coalition at their observed values and replacing all other features with values from a fixed reference input (Janzing et al., 2020; Lundberg et al., 2020). At each split, if the observed and reference values lead to the same branch, we always follow that branch. Otherwise, the branch depends on whether the feature belongs to the coalition. This gives a similarly sized tree-ensemble representation of the game, so we can apply Algorithm 1 directly to compute its Shapley variances.

3.3 GAUSSIAN PROCESS REPRESENTATION

A Gaussian process (GP) with a product-Hamming kernel gives a surrogate whose Shapley moments can be computed exactly. For center coalitions $T_1, \ldots, T_r \subseteq [d]$, positive lengthscales ℓ_j, and fitted coefficients $c, \alpha_a \in \mathbb{R}$, its posterior mean is

$$\hat{f}(S) = c + \sum_{a=1}^{r} \alpha_a \exp\left[-\frac{1}{d} \sum_{j \in S \triangle T_a} \frac{1}{\ell_j}\right].$$

ShaplEIG uses a Hamming-kernel GP to estimate Shapley values and their posterior uncertainty (Rundel et al., 2026). Here we compute the variance of marginal contributions across coalitions for the fixed fitted game. Our moment computation uses the product-integral and Gauss–Legendre quadrature approach developed for Shapley values in QuadraSHAP (Mohammadi et al., 2026) and Quadrature-TreeSHAP (Wettenstein et al., 2026). For second moments, we apply this construction to pairs of kernel terms, whose products retain the same factorization across players.

Theorem 3.5 (GP Fitting and Exact Moment Computation)**.** *A dense GP on n distinct training coalitions uses $r = n$ centers. Fitting its lengthscales and coefficients costs $O(t(n^3 + n^2 d))$ operations for t objective-and-gradient evaluations. For any fixed fitted game with r centers, given $\lceil d/2 \rceil$ Gauss–Legendre nodes and weights, exact arithmetic computes all Shapley variances $\sigma_i^2(\hat{f})$ in $O(r^2 d^2)$ operations.*

Appendix A.3 gives the moment formulas and proof; quadrature setup adds $O(d^3)$ operations. Appendix B.2.3 describes fitting with fewer centers in the high budget regime.

4 ESTIMATING SHAPLEY VARIANCE

When an exact representation is unavailable or too large to evaluate, we estimate Shapley variance.

Table 1: Variance estimation on five applications at full FFD budgets: each estimator gives the 25th percentile, median, and 75th percentile of normalized mean squared error (nMSE); lower is better. Medals compare each statistic independently within each application. Each application has 64 dataset/model settings and sixteen instances per setting. We use compact scientific notation, e.g., $7.1e^{-5}$ represents 7.1×10^{-5}.

Application	σ^2SHAP [Polynomial]			σ^2SHAP [GP]			σ^2SHAP [Tree]			FFD			MC		
	25th	Median	75th	25th	Median	75th	25th	Median	75th	25th	Median	75th	25th	Median	75th
Local explanations	$7.1e^{-5}$	$7.3e^{-4}$	$3.5e^{-3}$	$4.4e^{-5}$	$8.9e^{-4}$	$3.5e^{-3}$	$1.1e^{-3}$	$3.2e^{-3}$	$9.4e^{-3}$	$3.7e^{-4}$	$2.7e^{-3}$	$1.2e^{-2}$	$5.2e^{-3}$	$1.2e^{-2}$	$2.5e^{-2}$
Feature retraining	$1.4e^{-2}$	$3.0e^{-2}$	$5.8e^{-2}$	$1.2e^{-2}$	$3.3e^{-2}$	$6.6e^{-2}$	$1.5e^{-2}$	$3.2e^{-2}$	$5.7e^{-2}$	$2.2e^{-2}$	$6.4e^{-2}$	$1.4e^{-1}$	$2.8e^{-2}$	$4.3e^{-2}$	$7.1e^{-2}$
Training group valuation	$4.9e^{-2}$	$8.5e^{-2}$	$1.2e^{-1}$	$3.7e^{-2}$	$6.1e^{-2}$	$9.8e^{-2}$	$4.9e^{-2}$	$8.9e^{-2}$	$1.3e^{-1}$	$1.5e^{-1}$	$5.2e^{-1}$	$9.7e^{-1}$	$5.5e^{-2}$	$9.3e^{-2}$	$1.3e^{-1}$
Ensemble selection	$8.0e^{-2}$	$1.5e^{-1}$	$1.9e^{-1}$	$6.0e^{-2}$	$1.1e^{-1}$	$1.6e^{-1}$	$8.2e^{-2}$	$1.5e^{-1}$	$2.0e^{-1}$	$6.1e^{-1}$	$2.0e^{0}$	$2.5e^{0}$	$8.2e^{-2}$	$1.5e^{-1}$	$1.9e^{-1}$
Transformer ablation	$1.3e^{-5}$	$1.5e^{-4}$	$8.0e^{-4}$	$7.4e^{-6}$	$8.1e^{-5}$	$4.6e^{-4}$	$2.5e^{-4}$	$2.0e^{-3}$	$7.2e^{-3}$	$1.1e^{-4}$	$1.4e^{-3}$	$7.5e^{-3}$	$6.3e^{-3}$	$1.6e^{-2}$	$3.2e^{-2}$

The primary estimator we propose, called σ^2SHAP, is a control variate with corrections that we instantiate with structured games whose Shapley variance we can compute exactly. We compare this method to the natural Monte Carlo estimator built from sampled permutations and a generalized version of the Fractional Factorial Design (FFD) estimator for Shapley values (Zhou et al., 2026). FFD estimates the first two marginal moments with a fixed $O(d^2)$ query budget; its construction and error bound are in Appendix B.3.

Table 1 compares the variance estimators at FFD's query budget; Section 6 describes the games and evaluation protocol. The polynomial estimator's median nMSE is 7.3×10^{-4} for local explanations, compared with MC's 1.2×10^{-2}; their medians are closer for training-group valuation (0.085 versus 0.093) and ensemble selection (both 0.15 at the displayed precision). Appendix D.2 defines each application, and Appendix E.2 compares error by query budget and computation time.

4.1 σ^2SHAP Estimators

The primary method we use is a control variate strategy, inspired by a similar method used in Shapley value and Shapley interaction estimation (Witter et al., 2025; Thies et al., 2026).

We first fit a surrogate $\hat{f}$ to f on samples. Minimizing prediction loss does not directly minimize Shapley-variance error, and $\sigma_i^2(\hat{f})$ can be far from $\sigma_i^2(f)$ even when $\hat{f}$ and f are close.

Fortunately, we can bypass this limitation by adding two correction terms. Observe that

$$\sigma_i^2(f) = \sigma_i^2(f - \hat{f} + \hat{f}) = \sigma_i^2(\hat{f}) + 2\,\mathrm{Cov}\Big(\Delta_i \hat{f}, \Delta_i(f - \hat{f})\Big) + \sigma_i^2(f - \hat{f}).$$

Because $\hat{f}$ is structured, we exactly compute its Shapley variance and then use the natural Monte Carlo estimator[1] for the last two terms. We use all samples of f both to fit the surrogate and estimate the correction terms. While it introduces a small bias, we find reusing samples reduces the error.

We describe how we instantiate the estimator below, and additional details are in Appendix B.

- **σ^2SHAP [Polynomial].** We fit a tree ensemble to find interactions, then keep its largest Fourier terms to make exact variance computation practical. Least-squares refitting adjusts the coefficients after smaller terms are discarded. Converting to Möbius monomials and refitting lets subset terms vary independently, freeing coefficients tied together by each Fourier term. Lemma 3.2 then gives the polynomial's exact Shapley variance. Appendix E.6 ablates the support selection, refitting steps, and residual correction.

[1]We use the usual sample covariance and sample variance, computed from at least two independent permutation draws.

- σ^2**SHAP [Tree].** We fit a tree ensemble $\hat{f}$ to learn joint effects of players; its leaf representation permits exact variance computation with Algorithm 1. As in Butler et al. (2025); Thies et al. (2026), the hyperparameters are optimized on a training set of games.
- σ^2**SHAP [GP].** We fit a product-Hamming Gaussian process to share information between coalitions with similar player membership; Theorem 3.5 gives the fitted mean's exact moments. We fix the intercept to the sampled mean during fitting, so predictions approach it as correlations with all kernel centers vanish. At larger budgets, a finite-rank GP with at most 128 centers limits fitting and moment costs. Pivoted Cholesky selects coalition patterns poorly represented by the existing centers.

5 Applications to Estimating Shapley Values

In the prior section, we used insights from Shapley value estimators to estimate the Shapley variance. Here, we use insights from the Shapley variance to explain when the *paired sampling* heuristic helps estimate the Shapley value.

Paired sampling involves picking a coalition S and its complement $S^c = [d] \setminus S$ together. This method is widely used to reduce estimation error by as much as orders of magnitude, but it was not understood why it worked so well (Covert & Lee, 2021; Mitchell et al., 2022). Recently, Fumagalli et al. (2026) explained how differences between paired values isolate the *odd* component of a game:

$$f_{\text{odd}}(S) = \frac{f(S) - f(S^c)}{2}, \qquad f_{\text{even}}(S) = \frac{f(S) + f(S^c)}{2}. \tag{7}$$

We have $f(S) = f_{\text{odd}}(S) + f_{\text{even}}(S)$. Further, the Shapley value depends only on the odd component, i.e., $\phi_i(f) = \phi_i(f_{\text{odd}})$.

Therefore sampling in pairs isolates the relevant component of the game for Shapley values. However, somewhat surprisingly, paired sampling does not *always* improve Shapley value estimators.[2] In this section, we characterize exactly when paired sampling helps the Monte Carlo and weighted least-squares estimators.

Figure 2 compares the benefit of pairing with the game's even-to-odd variance ratio. The relationship differs across estimators: Monte Carlo error depends directly on marginal-contribution variance, while regression error also depends on the sampled fit. We derive the corresponding decompositions to identify when pairing helps.

Writing $\boldsymbol{\phi} = (\phi_i(f))_{i=1}^d$, we measure an estimator's expected squared error by $\mathbb{E}\|\boldsymbol{\phi} - \hat{\boldsymbol{\phi}}\|_2^2 = \sum_{i=1}^d \mathbb{E}[(\phi_i(f) - \hat{\phi}_i(f))^2]$, where the expectation is over the estimator's random samples.

5.1 Monte Carlo Shapley Value Estimators

For each player i, draw contexts $S_j \subseteq [d] \setminus \{i\}$ independently from the Shapley distribution and write $S_j^{c_i} = ([d] \setminus \{i\}) \setminus S_j$. With an even number m of marginal contributions per player, the unpaired and paired MC estimators are

$$\hat{\phi}_i^{\text{unpaired}} = \frac{1}{m}\sum_{j=1}^{m} \Delta_i f(S_j), \qquad \hat{\phi}_i^{\text{paired}} = \frac{1}{m}\sum_{j=1}^{m/2} \big(\Delta_i f(S_j) + \Delta_i f(S_j^{c_i})\big).$$

Notice that the paired estimator uses $m/2$ independent base contexts and their complements.

Using the independence of samples, it is easy to see that the expected squared error of the Monte Carlo estimators depends on the Shapley variance.

[2] Li et al. (2026, Theorem 3.2) gives an exact criterion for when pairing helps a restricted class of maximum sample reuse estimators; unpaired, unadjusted versions can have high variance because they use raw game values (Witter et al., 2025; Fumagalli et al., 2026).

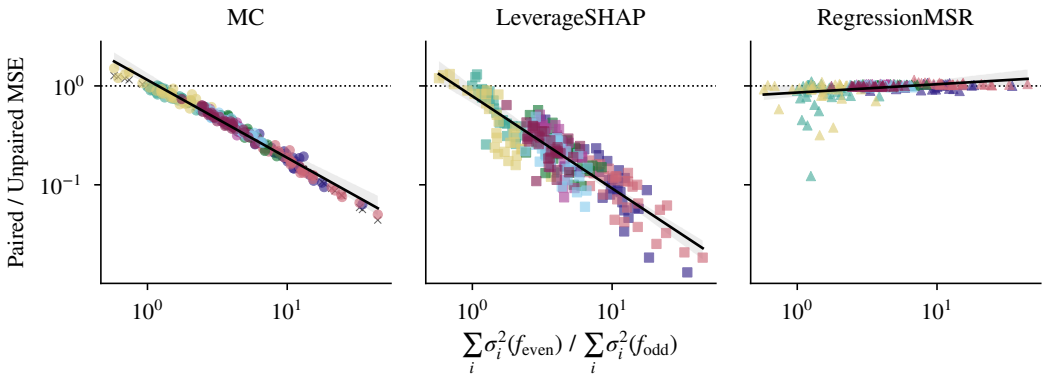

Figure 2: Pairing benefit at $64d$ queries; values below one favour pairing. Held-out R^2: MC 0.968, LeverageSHAP 0.774, RegressionMSR 0.034. Black crosses give the exact Monte Carlo prediction; bands show 95% family-bootstrap intervals.

Lemma 5.1 (Monte Carlo Error Decomposition; proof in Appendix C.2)**.** *At the same marginal-sample count m, the two estimators satisfy*

$$\mathbb{E}\|\boldsymbol{\phi} - \hat{\boldsymbol{\phi}}^{\text{unpaired}}\|_2^2 = \frac{1}{m}\sum_{i=1}^{d}\big(\sigma_i^2(f_{\text{even}}) + \sigma_i^2(f_{\text{odd}})\big),$$

$$\mathbb{E}\|\boldsymbol{\phi} - \hat{\boldsymbol{\phi}}^{\text{paired}}\|_2^2 = \frac{2}{m}\sum_{i=1}^{d}\sigma_i^2(f_{\text{odd}}).$$

Pairing removes the even component but halves the number of independent contexts, doubling the contribution of the remaining odd variance. Hence:

Corollary 5.2. *The paired Monte Carlo estimator has strictly lower error exactly when*

$$\sum_i \sigma_i^2(f_{\text{odd}}) < \sum_i \sigma_i^2(f_{\text{even}}) \tag{8}$$

Stratification. Coalition-size stratification can reduce MC variance (Castro et al., 2017; Wu et al., 2023); Appendix C.2 gives its combination with the pairing decomposition.

5.2 Regression Shapley Value Estimators

Regression estimators fit all Shapley values jointly, so their error depends on the game's residual from the Shapley additive approximation. For $d \geq 2$, draw $n \geq d-1$ nonempty proper coalitions independently, giving each coalition positive sampling probability equal to that of its complement. Let $\hat{\boldsymbol{\phi}}$ be the player coefficients of the weighted least-squares additive fit constrained to match the exact empty- and full-coalition values. The weights are the Shapley regression weights divided by the coalition sampling probabilities. The full regression setup is in Appendix C.3.

Let $\mathbf{W}_n \in \mathbb{R}^{d\times n}$ be the random linear map from sampled residuals to coefficient errors in this unpaired regression fit, defined in Equation equation 15 in Appendix C.3; it depends only on the sampled coalitions and their weights. Writing the sampled coalitions as $S_1, \ldots, S_n$ and conditioning on a full-rank design at each budget, the expected squared Shapley-vector error for $p \in \{\text{even}, \text{odd}\}$ is

$$R_p(n) = \mathbb{E}\left\|\sum_{j=1}^{n}[\mathbf{W}_n]_{:,j}\left[f_p(S_j) - \sum_{i\in S_j}\phi_i(f_p) - f_p(\emptyset)\right]\right\|_2^2.$$

Theorem 5.3 (Regression Error Decomposition; proof in Appendix C.3)**.** *At the same coalition-row budget n, unpaired regression and paired regression using $n/2$ independent base rows and their*

complements satisfy

$$\mathbb{E}\|\boldsymbol{\phi} - \hat{\boldsymbol{\phi}}^{\text{unpaired}}\|_2^2 = R_{\text{even}}(n) + R_{\text{odd}}(n),$$
$$\mathbb{E}\|\boldsymbol{\phi} - \hat{\boldsymbol{\phi}}^{\text{paired}}\|_2^2 = R_{\text{odd}}(n/2).$$

Pairing requires n even and $n/2 \geq d - 1$.

Pairing removes the even component but halves the number of independent rows; this need not double the odd error. Hence:

Corollary 5.4. *Under the assumptions of Theorem 5.3, the paired regression estimator has strictly lower expected squared error exactly when*

$$R_{\text{even}}(n) > R_{\text{odd}}(n/2) - R_{\text{odd}}(n) \tag{9}$$

The odd term contains only residual error after subtracting the Shapley additive approximation. Since $\phi_i(f_{\text{even}}) = 0$, the even term retains the centered even signal; we therefore expect a smaller odd term when the additive approximation captures most of the odd component. The variance ratio in Figure 2 is a diagnostic; regression error also depends on how the sampled coalitions determine the fit (Appendix D.5).

Additive games. If f is additive, $f(S) - [f([d]) + f(\emptyset)]/2$ is odd and its Shapley additive approximation is exact. Both residuals vanish, so $R_{\text{even}}(n) = R_{\text{odd}}(n) = 0$: both fits recover the Shapley values exactly whenever their designs have full rank.

Stratification. Unlike the Monte Carlo estimator, size stratification can increase regression variance; Appendix C.3 gives a three-player example with a 25% increase.

6 Experiments

We compare 5,120 games across local explanations, feature retraining, training-group valuation, ensemble selection, and transformer ablation, with 64 dataset/model settings per application and sixteen instances per setting. Appendix D.2 describes the constructions.

Shapley value estimators and pairing. In the 240-game local-explanation comparison, the five leading Shapley value estimators use pairing (Table 4); Figure 2 and Section 5 explain its estimator-dependent benefit.

Shapley variance estimators ablation. Removing residual correction worsens all eight family medians for matched tree and polynomial surrogates (Table 5). Fourier selection and refitting followed by Möbius refitting improves on corrected Fourier-only fitting in seven families; selecting Möbius coefficients directly from trees worsens all eight. Table 1 and Figures 4 and 5 show that gains weaken on training-group and ensemble games; Appendix E examines FFD's sensitivity to higher-order structure and the Gaussian process predicts incorrect labels.

Explanation distributions. Figure 1 and Appendix F compare marginal distributions and value-only versus variance-enhanced explanations across tabular, image, and text inputs. Similar means can hide different context dependence, exposed by the variance display while preserving mean attributions.

Fourier structure. Figure 3 compares Fourier magnitudes by degree for $\Delta_i f$ and $(\Delta_i f)^2$.

Limitations and Future Work

Our analysis takes the coalition game as given and does not resolve difficulties in choosing a value function that represents the intended explanation (Janzing et al., 2020; Michiels et al., 2024; Fumagalli et al., 2025). While Shapley variance gives more information than the Shapley value alone, it does not specify the full marginal-contribution distribution. Future work could explore other summaries of this distribution. Higher-order moments such as skewness and kurtosis can tease apart more subtle distributional patterns and characterize the performance of Shapley variance estimators. Another complementary measure is the *signed Shapley value*, which separates the expected positive and negative marginal contributions. Summary statistics like the median are less sensitive to extreme marginal contributions and may be useful in downstream applications. Appendix G establishes hardness results for exact signed moments, cumulative probabilities, and quantiles in tree ensembles.

AI USE STATEMENT

The authors developed the research questions and core conceptual framework and directed the methodology and interpretation. Generative AI assisted with proof writing, literature review, code implementation, figure preparation, and prose polishing. The authors take responsibility for the final content.

REPRODUCIBILITY STATEMENT

Assumptions accompany the formal results in the main text and appendix; proofs appear in Appendices A, B, C, and G. Section 6 summarizes the experiments, with data processing, game construction, estimator settings, and evaluation protocols in Appendices D and F. The supplementary code contains implementations, tests, numerical results, and scripts for regenerating the reported figures and tables from supplied numerical inputs. Its README and benchmark-data documentation specify the environment, commands, synthetic games available for fresh estimator runs, and reproduction scope, including the omission of trained model snapshots and full game caches needed for complete benchmark reruns.

REFERENCES

John F. Banzhaf, III. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19(2):317–343, 1965. URL https://cir.nii.ac.jp/crid/1570009750969875328.

Robert Bartlett, Adair Morse, Richard Stanton, and Nancy Wallace. Consumer-lending discrimination in the FinTech era. *Journal of Financial Economics*, 143(1):30–56, 2022. doi: 10.1016/j.jfineco.2021.05.047. URL https://doi.org/10.1016/j.jfineco.2021.05.047.

Blair Bilodeau, Natasha Jaques, Pang Wei Koh, and Been Kim. Impossibility theorems for feature attribution. *Proceedings of the National Academy of Sciences*, 121(2):e2304406120, 2024. doi: 10.1073/pnas.2304406120. URL https://doi.org/10.1073/pnas.2304406120.

Landon Butler, Abhineet Agarwal, Justin Singh Kang, Yigit Efe Erginbas, Bin Yu, and Kannan Ramchandran. ProxySPEX: Inference-efficient interpretability via sparse feature interactions in LLMs. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2426. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/68dced1638d71d4d21598f79ae91262e-Abstract-Conference.html.

Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the Shapley value based on sampling. *Computers & Operations Research*, 36(5):1726–1730, 2009. doi: 10.1016/j.cor.2008.04.004. URL https://doi.org/10.1016/j.cor.2008.04.004.

Javier Castro, Daniel Gómez, Elisenda Molina, and Juan Tejada. Improving polynomial estimation of the Shapley value by stratified random sampling with optimum allocation. *Computers & Operations Research*, 82:180–188, 2017. doi: 10.1016/j.cor.2017.01.019. URL https://doi.org/10.1016/j.cor.2017.01.019.

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. ACM, 2016. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

Ian Covert and Su-In Lee. Improving KernelSHAP: Practical Shapley value estimation using linear regression. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 3457–3465. PMLR, 2021. URL https://proceedings.mlr.press/v130/covert21a.html.

Dean De Cock. Ames, Iowa: Alternative to the Boston housing data as an end of semester regression project. *Journal of Statistics Education*, 19(3), 2011. doi: 10.1080/10691898.2011.11889627. URL https://jse.amstat.org/v19n3/decock.pdf.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Desai, David Salinas, and Frank Hutter. TabArena: A living benchmark for machine learning on tabular data. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-0519. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/1697e3fb412da11dc9488249f9e7bbc9-Paper-Datasets_and_Benchmarks_Track.pdf. Datasets and Benchmarks Track.

Shaheen S. Fatima, Michael Wooldridge, and Nicholas R. Jennings. An analysis of the Shapley value and its uncertainty for the voting game. In *Agent-Mediated Electronic Commerce. Designing Trading Agents and Mechanisms*, volume 3937 of *Lecture Notes in Computer Science*, pp. 85–98. Springer, 2006. doi: 10.1007/11888727_7. URL https://www.cs.ox.ac.uk/people/michael.wooldridge/pubs/amec05b.pdf. AMEC 2005 workshop; revised selected papers published in 2006.

Luca Franceschi, Michele Donini, Cédric Archambeau, and Matthias Seeger. Explaining probabilistic models with distributional values. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 13840–13863, 2024. URL https://proceedings.mlr.press/v235/franceschi24a.html.

Daniel Vidali Fryer, Inga Strümke, and Hien D. Nguyen. Shapley values for feature selection: The good, the bad, and the axioms. *IEEE Access*, 9:144352–144360, 2021. doi: 10.1109/ACCESS.2021.3119110. URL https://doi.org/10.1109/ACCESS.2021.3119110.

Fabian Fumagalli, Maximilian Muschalik, Eyke Hüllermeier, Barbara Hammer, and Julia Herbinger. Unifying feature-based explanations with functional ANOVA and cooperative game theory. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258, pp. 5140–5148. PMLR, 2025. URL https://proceedings.mlr.press/v258/fumagalli25a.html.

Fabian Fumagalli, Landon Butler, Justin Singh Kang, Kannan Ramchandran, and R. Teal Witter. An odd estimator for Shapley values. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=xILwgiWAUk.

Amirata Ghorbani and James Y. Zou. Data Shapley: Equitable valuation of data for machine learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2242–2251. PMLR, 2019. URL http://proceedings.mlr.press/v97/ghorbani19c.html.

Amirata Ghorbani and James Y. Zou. Neuron Shapley: Discovering the responsible neurons. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/41c542dfe6e4fc3deb251d64cf6ed2e4-Abstract.html.

Jeremy Goldwasser and Giles Hooker. Stabilizing estimates of Shapley values with control variates. In *Explainable Artificial Intelligence*, pp. 416–439. Springer, 2024. doi: 10.1007/978-3-031-63797-1_21. URL https://arxiv.org/abs/2310.07672.

Ali Gorji, Andisheh Amrollahi, and Andreas Krause. SHAP values via sparse Fourier representation. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/1b331c20064e37e204a5bcd12481bfac-Abstract-Conference.html.

Michel Grabisch and Marc Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of Game Theory*, 28(4):547–565, 1999. doi: 10.1007/s001820050125. URL https://doi.org/10.1007/s001820050125.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin Hoo, Magnus Bühler, Anurag Garg, Dominik Safaric, Jake Robertson, Benjamin Jäger, Simone Alessi, Adrian Hayler, Vladyslav Moroshan, Lennart Purucker, Philipp Singer, Alan Arazi, Julien Siems, Jan Hendrik Metzen, Georg Grab, Nick Erickson, Siyuan Guo, Eliott Kalfon, Simon Bing, David Salinas, Clara Cornu, Lilly Charlotte Wehrhahn, Diana Kriuchkova, Kursat Kaya, Lydia Sidhoum, Marie Salmon, Jerry Chen, Madelon Hulsebos, Yann LeCun, Samuel Müller, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. TabPFN-3: Technical report, 2026. URL https://arxiv.org/abs/2605.13986.

Rohan Gupta, Iván Arcuschin, Thomas Kwa, and Adrià Garriga-Alonso. InterpBench: Semi-synthetic transformers for evaluating mechanistic interpretability techniques. In *Advances in Neural Information Processing Systems*, volume 37, pp. 92922–92951. Curran Associates, Inc., 2024. doi: 10.52202/079017-2950. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/a8f7d43ae092d9a5295775eb17f3f4f7-Paper-Datasets_and_Benchmarks_Track.pdf. Datasets and Benchmarks Track.

Raoul Heese, Sascha Mücke, Matthias Jakobs, Thore Gerlach, and Nico Piatkowski. Shapley values with uncertain value functions. In *Advances in Intelligent Data Analysis XXI*, volume 13876 of *Lecture Notes in Computer Science*, pp. 156–168. Springer, 2023. doi: 10.1007/978-3-031-30047-9_13. URL https://arxiv.org/abs/2301.08086.

Raoul Heese, Thore Gerlach, Sascha Mücke, Sabine Müller, Matthias Jakobs, and Nico Piatkowski. Explaining quantum circuits with Shapley values: towards explainable quantum machine learning. *Quantum Machine Intelligence*, 7:27, 2025. doi: 10.1007/s42484-025-00254-8. URL https://doi.org/10.1007/s42484-025-00254-8.

Xuanxiang Huang and Joao Marques-Silva. On the failings of Shapley values for explainability. *International Journal of Approximate Reasoning*, 171:109112, 2024. doi: 10.1016/j.ijar.2023.109112. URL https://doi.org/10.1016/j.ijar.2023.109112.

Kosuke Imai, Zhichao Jiang, D. James Greiner, Ryan Halen, and Sooahn Shin. Experimental evaluation of algorithm-assisted human decision-making: Application to pretrial public safety assessment. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 186(2):167–189, 2023. doi: 10.1093/jrsssa/qnad010. URL https://imai.fas.harvard.edu/research/prai/.

Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. Feature relevance quantification in explainable AI: A causal problem. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2907–2916. PMLR, 2020. URL http://proceedings.mlr.press/v108/janzing20a.html.

Justin Singh Kang, Landon Butler, Abhineet Agarwal, Yigit Efe Erginbas, Ramtin Pedarsani, Bin Yu, and Kannan Ramchandran. SPEX: Scaling feature interaction explanations for LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 28878–28903. PMLR, 2025. URL https://proceedings.mlr.press/v267/kang25a.html.

Vladislav Kargin. Uncertainty of the Shapley value. *International Game Theory Review*, 7(4):517–529, 2005. doi: 10.1142/S0219198905000685. URL https://doi.org/10.1142/S0219198905000685.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, pp. 3146–3154. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.

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. doi: 10.1609/aaai.v38i12.29225. URL https://ojs.aaai.org/index.php/AAAI/article/view/29225. Issue 12.

Indra Kumar, Carlos Scheidegger, Suresh Venkatasubramanian, and Sorelle A. Friedler. Shapley residuals: Quantifying the limits of the Shapley value for explanations. In *Advances in Neural Information Processing Systems*, volume 34, pp. 26598–26608, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/dfc6aa246e88ab3e32caeaaecf433550-Abstract.html.

Yongchan Kwon and James Zou. Beta Shapley: a unified and noise-reduced data valuation framework for machine learning. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 8780–8802. PMLR, 2022. URL https://proceedings.mlr.press/v151/kwon22a.html.

Weida Li and Yaoliang Yu. Robust data valuation with weighted Banzhaf values. In *Advances in Neural Information Processing Systems 36*, 2023. doi: 10.52202/075280-2637. URL https://papers.nips.cc/paper_files/paper/2023/hash/bdb0596d13cfccf2db6f0cc5280d2a3f-Abstract-Conference.html.

Weida Li, Yaoliang Yu, and Bryan Kian Hsiang Low. Adalina: Adaptive linear approximation for the Shapley value and beyond. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=hMTCuqjCW8.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4765–4774, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

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. doi: 10.1038/s42256-019-0138-9. URL https://doi.org/10.1038/s42256-019-0138-9.

Joran Michiels, Johan Suykens, and Maarten De Vos. Explaining the model and feature dependencies by decomposition of the Shapley value. *Decision Support Systems*, 182:114234, 2024. doi: 10.1016/j.dss.2024.114234. URL https://doi.org/10.1016/j.dss.2024.114234.

Rory Mitchell, Joshua Cooper, Eibe Frank, and Geoffrey Holmes. Sampling permutations for Shapley value estimation. *Journal of Machine Learning Research*, 23(43):1–46, 2022. URL https://jmlr.org/papers/v23/21-0439.html.

Majid Mohammadi, Grigory Reznikov, Pavel Sinitcyn, Krikamol Muandet, and Siu Lun Chau. QuadraSHAP: Stable and scalable Shapley values for product games via Gauss-Legendre quadrature, 2026. URL https://arxiv.org/abs/2605.05870v3. arXiv preprint, version 3.

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

Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019. doi: 10.1126/science.aax2342. URL https://doi.org/10.1126/science.aax2342.

Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. doi: 10.1017/CBO9781139814782. URL https://doi.org/10.1017/CBO9781139814782.

Ramin Okhrati and Aldo Lipani. A multilinear sampling algorithm to estimate Shapley values. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 7992–7999. IEEE, 2021. doi: 10.1109/ICPR48806.2021.9412511. URL https://doi.org/10.1109/ICPR48806.2021.9412511.

Karl Michael Ortmann. The link between the Shapley value and the beta factor. *Decisions in Economics and Finance*, 39:311–325, 2016. doi: 10.1007/s10203-016-0178-0. URL https://doi.org/10.1007/s10203-016-0178-0.

Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(65):1939–1959, 2005. URL https://jmlr.org/papers/v6/quinonero-candela05a.html.

Benedek Rozemberczki, Lauren Watson, Péter Bayer, Hao-Tsung Yang, Olivér Kiss, Sebastian Nilsson, and Rik Sarkar. The Shapley value in machine learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 5572–5579, 2022. doi: 10.24963/ijcai.2022/778. URL https://doi.org/10.24963/ijcai.2022/778.

David Rundel, Fabian Fumagalli, Maximilian Muschalik, Bernd Bischl, and Matthias Feurer. ShaplEIG: Bayesian experimental design for Shapley value estimation. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ub9PwBtHqD.

Lloyd S. Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker (eds.), *Contributions to the Theory of Games (AM-28), Volume II*, pp. 307–317. Princeton University Press, 1953. doi: 10.1515/9781400881970-018. URL https://doi.org/10.1515/9781400881970-018.

Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. The Shapley Taylor interaction index. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 9259–9268. PMLR, 2020. URL https://proceedings.mlr.press/v119/sundararajan20a.html.

Santo M. A. R. Thies, Hubert Baniecki, R. Teal Witter, Eyke Hüllermeier, Maximilian Muschalik, and Fabian Fumagalli. Proxy-based approximation of Shapley and Banzhaf interactions, 2026. URL https://arxiv.org/abs/2605.22738. arXiv preprint arXiv:2605.22738.

Che-Ping Tsai, Chih-Kuan Yeh, and Pradeep Ravikumar. Faith-Shap: The faithful Shapley interaction index. *Journal of Machine Learning Research*, 24(94):1–42, 2023. URL https://jmlr.org/papers/v24/22-0202.html.

Robert James Weber. Probabilistic values for games. In Alvin E. Roth (ed.), *The Shapley Value*, pp. 101–120. Cambridge University Press, 1988. doi: 10.1017/cbo9780511528446.008. URL https://doi.org/10.1017/cbo9780511528446.008.

Ron Wettenstein, Rory Mitchell, and Peng Yu. Quadrature-TreeSHAP: Depth-independent TreeSHAP and Shapley interactions, 2026. URL https://arxiv.org/abs/2605.04497. arXiv preprint.

R. Teal Witter, Yurong Liu, and Christopher Musco. Regression-adjusted Monte Carlo estimators for Shapley values and probabilistic values. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0715. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/1e77af93008ee6cd248a31723ce357d8-Abstract-Conference.html.

Mengmeng Wu, Ruoxi Jia, Changle Lin, Wei Huang, and Xiangyu Chang. Variance reduced Shapley value estimation for trustworthy data valuation. *Computers & Operations Research*, 159:106305, 2023. doi: 10.1016/j.cor.2023.106305. URL https://doi.org/10.1016/j.cor.2023.106305.

Zheng Zhou, Robert Mee, Herbert Hamers, and Wei Zheng. Fast approximation of Shapley values through fractional factorial designs. *Journal of the American Statistical Association*, 121(553): 525–535, 2026. doi: 10.1080/01621459.2025.2529027. URL https://doi.org/10.1080/01621459.2025.2529027.

APPENDIX CONTENTS

A Properties and Exact Computation **17**
- A.1 Variance Representations . . . 17
- A.2 Exact Tree Computation . . . 17
- A.3 Exact Gaussian Process Moments . . . 18
- A.4 Equal Residual Norms with Different Shapley Variances . . . 19

B Estimating Shapley Variance **20**
- B.1 Monte Carlo Variance Estimation . . . 20
- B.2 Surrogate Estimators . . . 20
- B.3 Fractional Factorial Design . . . 23

C Estimating Shapley Values **25**
- C.1 Size and Parity Decompositions . . . 25
- C.2 Monte Carlo Shapley Error . . . 25
- C.3 Regression Identities . . . 26

D Experimental Setup **28**
- D.1 Games and Exact References . . . 28
- D.2 Application Games . . . 28
- D.3 Budgets and Evaluation . . . 29
- D.4 Surrogate Settings . . . 30
- D.5 Shapley Value Evaluation . . . 31

E Additional Results **32**
- E.1 Original Local-Explanation Comparison . . . 32
- E.2 Query and Computation Budgets . . . 32
- E.3 Higher-Order Structure and Transfer . . . 34
- E.4 GP Failure Analysis . . . 35
- E.5 Shapley Value Estimator Comparison . . . 35
- E.6 Variance Estimator Ablations . . . 37

F Explanation Examples **38**
- F.1 Figure 1 Protocol . . . 38
- F.2 Text Explanation . . . 38

G Limits of Computation **40**
- G.1 Value-Oracle Approximation Lower Bound . . . 40
- G.2 Signed Marginal Moments . . . 40
- G.3 CDF and Quantile Hardness . . . 41

A Properties and Exact Computation

Since every context has positive Shapley probability, variance vanishes exactly when the marginal contribution is constant:

$$\sigma_i^2(f) = 0 \iff \Delta_i f(S) = \phi_i(f) \quad \text{for all } S \subseteq [d] \setminus \{i\},$$
$$\sigma_i^2(f) + \phi_i(f)^2 = 0 \iff \Delta_i f(S) = 0 \quad \text{for all } S \subseteq [d] \setminus \{i\}.$$

Thus the value and variance together identify null players.

A.1 Variance Representations

Möbius representation (Lemma 3.2). Write $f(S) = \sum_{T \subseteq S} m(T)$, where $m(T) = \sum_{U \subseteq T} (-1)^{|T|-|U|} f(U)$. For a Shapley context S outside player i,

$$\Delta_i f(S) = \sum_{T \ni i} m(T) \mathbf{1}[T \setminus \{i\} \subseteq S].$$

In a uniform permutation, all players of $U \subseteq [d] \setminus \{i\}$ precede i exactly when i is last among $U \cup \{i\}$, which has probability $1/(|U|+1)$. Consequently,

$$\mathbb{E}[\Delta_i f] = \sum_{T \ni i} \frac{m(T)}{|T|},$$
$$\mathbb{E}[(\Delta_i f)^2] = \sum_{T, T' \ni i} \frac{m(T) m(T')}{|T \cup T'|}.$$

Subtracting the squared mean gives equation 6.

Fourier representation (Lemma 3.1). The Shapley law on contexts outside i is the mixture that draws p uniformly from $[0,1]$ and then includes each other player independently with probability p:

$$\Pr(S) = \int_0^1 p^{|S|}(1-p)^{d-1-|S|}\, dp = \frac{|S|!(d-1-|S|)!}{d!}.$$

Hence, for $U \subseteq [d] \setminus \{i\}$,

$$\mathbb{E}[\chi_U(S)] = \int_0^1 (2p-1)^{|U|}\, dp = \frac{\mathbf{1}[|U| \text{ even}]}{|U|+1}.$$

Taking a marginal of the Fourier expansion gives

$$\Delta_i f(S) = 2 \sum_{T \ni i} F(T) \chi_{T \setminus \{i\}}(S).$$

Products of characters satisfy $\chi_U \chi_V = \chi_{U \triangle V}$, so

$$\mathbb{E}[\Delta_i f] = 2 \sum_{\substack{T \ni i \\ |T| \text{ odd}}} \frac{F(T)}{|T|},$$
$$\mathbb{E}[(\Delta_i f)^2] = 4 \sum_{\substack{T, T' \ni i \\ |T| \equiv |T'| \pmod 2}} \frac{F(T) F(T')}{|T \triangle T'| + 1}.$$

Subtracting the squared mean proves equation 5. With r nonzero coefficients, processing the sets and shared players of each ordered pair takes $O(d)$ time, proving Corollary 3.3.

A.2 Exact Tree Computation

Proof of Lemma 3.4. A leaf's path indicator is $\prod_{j \in P} \frac{1+s_j z_j}{2}$. Expanding this product gives coefficient $2^{-|P|} \prod_{j \in T} s_j$ on each $T \subseteq P$. Multiplying by the leaf value and summing leaves proves Algorithm 1. There are at most $L2^h$ generated terms; forming and merging each term costs $O(d)$ using a trie of length-d indicator keys. Applying Corollary 3.3 to the merged coefficients costs $O(dr^2)$. For a finite background average, average the background-specific coefficient maps before applying the quadratic variance formula; averaging their variances would omit covariance.

A.3 Exact Gaussian Process Moments

For the center coalitions in Section 3.3, write $z_{aj} = \mathbf{1}[j \in T_a]$, $\rho_j = \exp[-1/(d\ell_j)]$, and $k_j(u, v) = \rho_j^{\mathbf{1}[u \neq v]}$. We can write the posterior mean as

$$\hat{f}(S) = c + \sum_{a=1}^{r} \alpha_a \prod_{j=1}^{d} k_j(\mathbf{1}[j \in S], z_{aj}).$$

The coefficients α_a absorb the fitted kernel output variance; both c and α_a are expressed in the game's original output units. We compute moments of the fixed fitted game, not posterior uncertainty about its Shapley values.

Fix player i and define $\beta_{ai} = \alpha_a(2z_{ai} - 1)(1 - \rho_i)$. The constant cancels from the marginal contribution:

$$\Delta_i \hat{f}(S) = \sum_{a=1}^{r} \beta_{ai} \prod_{j \neq i} k_j(\mathbf{1}[j \in S], z_{aj}).$$

Under the independent-inclusion mixture from Appendix A.1, conditioning on the inclusion probability t lets us factor both moments over players:

$$\phi_i(\hat{f}) = \sum_{a=1}^{r} \beta_{ai} \int_0^1 \prod_{j \neq i} \left[(1-t)k_j(0, z_{aj}) + tk_j(1, z_{aj})\right] dt, \tag{10}$$

$$\mathbb{E}_S[(\Delta_i \hat{f}(S))^2] = \sum_{a,b=1}^{r} \beta_{ai}\beta_{bi} \int_0^1 \prod_{j \neq i} \left[(1-t)k_j(0, z_{aj})k_j(0, z_{bj}) + tk_j(1, z_{aj})k_j(1, z_{bj})\right] dt. \tag{11}$$

Subtracting $\phi_i(\hat{f})^2$ from the second moment gives $\sigma_i^2(\hat{f})$. The two kernel sections in Equation 11 share the same coalition, so every factor remains linear in t.

Proof of Theorem 3.5. For the dense fit, evaluating the objective and gradient requires forming the kernel and all lengthscale derivatives in $O(n^2 d)$ operations. The shared Cholesky factorization and inverse solves cost $O(n^3)$; the derivatives reuse these solves. The initial evaluation and final coefficient refit add a constant number of these computations, giving $O(t(n^3 + n^2 d))$ operations for $t \geq 1$. Here t counts evaluations, since a line search can use several per optimizer iteration; the bound does not guarantee a global optimum.

Each integrand in Equations 10 and 11 is a polynomial of degree at most $d - 1$, while a Q-node Gauss–Legendre rule integrates every polynomial through degree $2Q - 1$ exactly. Thus $Q = \lceil d/2 \rceil$ nodes suffice in exact arithmetic. This is the product-game exactness argument of QuadraSHAP (Mohammadi et al., 2026, Section 3); Quadrature-TreeSHAP gives the corresponding degree-based rule for tree values and interactions (Wettenstein et al., 2026, Section 6). For each node and section, or pair of sections, prefix and suffix products compute every product excluding one player in $O(d)$ operations. There are r sections, r^2 section pairs, and $O(d)$ nodes, giving $O(rd^2)$ and $O(r^2 d^2)$ operations, respectively. □

With blocks of at most b kernel sections per side, moment computation requires $O(rd + b^2 d + d^2)$ working memory, including the fitted sections and conditional means at all quadrature nodes. Accumulating pair blocks avoids storing all $r^2 d$ player-specific pair contributions. Our implementation prepares the quadrature rule with a dense eigensolve costing an additional $O(d^3)$ per moment computation. For the full GP, $r = n$: second-moment computation is quadratic in the number of distinct training coalitions, whereas fitting includes a cubic cost per objective evaluation. The exactness guarantee concerns the fixed fitted game's moments; fitting an approximation of another game introduces separate error. Appendix B.2.3 gives the fitting and numerical implementation; Appendix D.4 gives the frozen settings and benchmark protocols.

A.4 Equal Residual Norms with Different Shapley Variances

For $S \subseteq [3]$, let $z_j(S) = 2\mathbf{1}\{j \in S\} - 1$ and define the two games

$$f_\pm(S) = z_1(S)z_2(S) \pm z_1(S)z_3(S) - (1 \pm 1).$$

Both vanish at the empty coalition. In the context order $\emptyset, \{2\}, \{3\}, \{2,3\}$, player 1's marginal contributions are

$$(\Delta_1 f_+(S))_S = (-4, 0, 0, 4),$$
$$(\Delta_1 f_-(S))_S = (0, 4, -4, 0).$$

The Shapley probabilities $(1/3, 1/6, 1/6, 1/3)$ give

$$\phi_1(f_\pm) = 0, \qquad \sigma_1^2(f_+) = \frac{32}{3}, \qquad \sigma_1^2(f_-) = \frac{16}{3}.$$

For the residual construction of Kumar et al. (2021), let $\mathbf{B} \in \mathbb{R}^{12\times 8}$ be the cube's incidence matrix, oriented toward larger coalitions. Let $\mathbf{B}_1 f$ retain only edges adding player 1, setting other entries to zero. The residual comes from fitting $\mathbf{B}h$ to $\mathbf{B}_1 f$ by unweighted least squares. For each $j \in \{2,3\}$, direct differencing gives

$$\mathbf{B}^\top \mathbf{B}_1(z_1 z_j) = 2z_1 z_j = \tfrac{1}{2}\mathbf{B}^\top \mathbf{B}(z_1 z_j).$$

Thus $h = f_\pm/2$ satisfies the normal equations, giving $\mathbf{r}_1^\pm = \mathbf{B}_1 f_\pm - \frac{1}{2}\mathbf{B} f_\pm$. The squared edge differences sum to 32, 16, and 16 in the three coordinates for both games. Orthogonality of the fitted gradient and residual therefore gives

$$\begin{aligned} \|\mathbf{r}_1^\pm\|_2^2 &= \|\mathbf{B}_1 f_\pm\|_2^2 - \tfrac{1}{4}\|\mathbf{B} f_\pm\|_2^2 \\ &= 32 - \tfrac{1}{4}(32 + 16 + 16) = 16, \\ \frac{\|\mathbf{r}_1^\pm\|_2}{\|\mathbf{B} f_\pm\|_2} &= \frac{4}{8} = \frac{1}{2}. \end{aligned}$$

The raw and normalized residual norms agree, while the Shapley variances differ by a factor of two.

B Estimating Shapley Variance

B.1 Monte Carlo Variance Estimation

When an exact representation is unavailable or too large, estimate the variance from sampled marginal contributions.

Algorithm 2 Monte Carlo estimation of Shapley variance

Require: Game oracle f and number of draws $m \geq 2$.
1: Cache $f(\emptyset)$ and $f([d])$.
2: **for** $j = 1, \ldots, m$ **do**
3: Independently draw a uniform permutation; let $S_{i,j}$ be the players preceding i.
4: Evaluate its interior prefixes and set $X_{i,j} \leftarrow f(S_{i,j} \cup \{i\}) - f(S_{i,j})$ for every i.
5: $\bar{X}_i \leftarrow m^{-1} \sum_{j=1}^{m} X_{i,j}$ for every i.
6: **return** $\hat{\sigma}_i^2(f) = (m-1)^{-1} \sum_{j=1}^{m} (X_{i,j} - \bar{X}_i)^2$ for every i.

Lemma B.1 (Monte Carlo Variance Error)**.** *For $\sigma_i^2(f) > 0$, let $\kappa_i = \mathbb{E}[(\Delta_i f(S) - \phi_i(f))^4]/\sigma_i^4(f)$ be the marginal kurtosis. Algorithm 2 uses at most $2 + m(d-1)$ game evaluations and returns unbiased estimates satisfying*

$$\mathbb{E}\|\boldsymbol{\sigma}^2(f) - \hat{\boldsymbol{\sigma}}^2(f)\|_2^2 = \frac{1}{m} \sum_{i:\sigma_i^2(f)>0} \sigma_i^4(f) \left(\kappa_i - 1 + \frac{2}{m-1} \right).$$

Players with zero marginal variance are estimated exactly.

Proof. Fix a player and let $X_1, \ldots, X_m$ be its independent marginal observations, with variance $\sigma^2 = \sigma_i^2(f)$ and fourth central moment μ_4. The unbiased sample variance $\hat{\sigma}^2$ is the average of $H_{ab} = (X_a - X_b)^2/2$ over unordered sample pairs. Independence and expansion give

$$\mathrm{Var}(H_{12}) = \frac{\mu_4 + \sigma^4}{2}, \qquad \mathrm{Cov}(H_{12}, H_{13}) = \frac{\mu_4 - \sigma^4}{4}.$$

Disjoint pairs are independent, so counting pairs sharing one observation gives

$$\begin{aligned} \mathrm{Var}(\hat{\sigma}^2) &= \frac{2\,\mathrm{Var}(H_{12})}{m(m-1)} + \frac{4(m-2)\,\mathrm{Cov}(H_{12}, H_{13})}{m(m-1)} \\ &= \frac{1}{m} \left(\mu_4 - \frac{m-3}{m-1} \sigma^4 \right). \end{aligned}$$

For $m = 2$, the shared-pair count is zero and only the first term is needed. If $\sigma^2 > 0$, substitute $\mu_4 = \kappa\sigma^4$; if $\sigma^2 = 0$, all observations are constant. Summing coordinate errors proves Lemma B.1; sharing a permutation across players does not change this sum. Caching the two endpoints leaves at most $d - 1$ new evaluations per permutation. □

Pairing caveat. Averaging complementary marginal pairs before computing a sample variance removes the even-game component and therefore estimates only $\sigma_i^2(f_{\text{odd}})$. Keeping both observations preserves both components, but their dependence changes the usual sample variance's bias as well as its variance. For m observations in independent pairs with identical marginals, variance σ^2, and within-pair covariance c, the usual sample variance has expectation $\sigma^2 - c/(m-1)$. Its variance depends on joint fourth moments, so Lemma B.1 does not apply unchanged to paired draws.

B.2 Surrogate Estimators

All three estimators fit queried endpoints and permutation-chain interiors, including repeated coalitions, and reuse the same chains for the covariance and residual-variance correction in Section 4. No rows are held out; training reuse prevents a general unbiasedness claim, and corrected estimates retain their sign. If the fitted game interpolates every queried value, the sampled residuals and correction vanish.

B.2.1 POLYNOMIAL ESTIMATOR

Use the same permutation-chain sampling rule and fit one XGBOOST ensemble on all queried rows. Retain at most 256 nonconstant Fourier terms and refit those characters and an intercept by ordinary least squares on all queried rows. The total coefficient count is at most one tenth of the actual query-row count; remove the smallest retained tree coefficients until the selected design has full column rank. Center the nonconstant columns and responses before fitting slopes, then recover the intercept from their means. For a fixed support, adding a constant to the game changes only the intercept.

Expand the fitted Fourier polynomial into Möbius monomials. Writing $x_j = \mathbf{1}[j \in S]$, each retained character satisfies

$$\chi_T(S) = \prod_{j \in T} (2x_j - 1) = \sum_{U \subseteq T} 2^{|U|}(-1)^{|T \setminus U|} \prod_{j \in U} x_j.$$

Take the union of these monomials, including terms whose converted coefficients cancel, and refit their coefficients independently by unpenalized least squares on all the same queried rows. Always retain the intercept; if necessary, retain the largest converted coefficients until the total count is at most one tenth of the actual query-row count, then remove trailing terms until the design has full column rank. The initial limit of 256 nonconstant Fourier terms applies to the Fourier selection only. Compute the refitted polynomial's exact variance directly from its Möbius coefficients using Lemma 3.2, then apply the covariance and residual-variance correction using its monomial predictions on the queried chains. The reported benchmark results and timings used the algebraically equivalent conversion to Fourier coefficients for moment computation. Both least-squares stages use the same queried values and the single initial tree fit. The independent Möbius refit can change the available function space by introducing subset terms with free coefficients; converting coordinates without refitting leaves the represented game unchanged.

B.2.2 TREE ESTIMATOR

Fit an XGBOOST ensemble and compute its exact variance with Algorithm 1, then apply the shared residual correction.

B.2.3 GAUSSIAN PROCESS FITTING AND MOMENT COMPUTATION

Fit. σ^2SHAP [GP] uses the dense fit when the number of scheduled observation rows, including duplicates and endpoints, is at most $16d$, and the regularized finite-rank fit above that threshold. The choice of fit depends only on the observed row count and player count; if the chosen fit fails, we report the failure without switching fits. Let $\bar{y}$ and s_y be the mean and sample standard deviation of all scheduled responses, including repeated coalitions and endpoints, using scale one for constant responses. We fix the raw-response intercept to $c = \bar{y}$ throughout hyperparameter optimization and coefficient fitting, and standardize responses by subtracting $\bar{y}$ and dividing by s_y. For the n distinct observed coalitions $S_1, \ldots, S_n$, write $x_{aj} = \mathbf{1}[j \in S_a]$, let $\mathbf{y} \in \mathbb{R}^n$ be their standardized response averages, and let N_a count repetitions of S_a. Using k_j from Appendix A.3, the standardized GP has fixed mean zero and covariance

$$[\mathbf{K}]_{ab} = s \prod_{j=1}^{d} k_j(x_{aj}, x_{bj}), \qquad \mathbf{C} = \mathbf{K} + 10^{-6} \operatorname{diag}(N_1^{-1}, \ldots, N_n^{-1}),$$

with positive output variance s. The repetition weights preserve the full-row Gaussian likelihood up to a hyperparameter-independent constant. For fixed hyperparameters, its coefficients solve

$$\boldsymbol{\alpha}^{\mathrm{std}} = \mathbf{C}^{-1}\mathbf{y}.$$

The main-text coefficients are $c = \bar{y}$ and $\alpha_a = s_y s \alpha_a^{\mathrm{std}}$.

The lengthscale priors are independent $\log \ell_j \sim \mathcal{N}(\sqrt{2} + \frac{1}{2}\log d, 3)$, and the output-scale prior is $\log s \sim \mathcal{N}(0, 1)$. We minimize

$$\frac{1}{2}\mathbf{y}^\top \mathbf{C}^{-1}\mathbf{y} + \frac{1}{2}\log\det \mathbf{C} - \sum_{j=1}^{d} \log p(\ell_j) - \log p(s).$$

Optimization uses L-BFGS-B and analytic gradients in log coordinates, with coordinate bounds $[-14, 14]$, at most 100 iterations, relative objective tolerance 10^{-9}, and gradient tolerance 10^{-5}. Initial lengthscales and output variance are their prior modes. The prior densities are evaluated on the original positive parameters; changing optimization coordinates introduces no Jacobian into this maximum a posteriori objective. We retain the initial parameters if optimization returns a nonfinite or worse objective and reject unresolved Cholesky proposals. The fixed noise variance 10^{-6} regularizes fitting in standardized response units; it is excluded from fitted-game marginal moments. Dense fitting uses $O(n^2 + nd)$ memory; its runtime and exact-moment costs are given in Theorem 3.5.

Finite-rank fit and regularization. Above the $16d$ threshold, σ^2SHAP [GP] uses at most $r = 128$ observed coalitions as kernel centers. The Nyström kernel defines a finite-dimensional GP whose coefficients are conditioned on all observations (Quiñonero-Candela & Rasmussen, 2005). For the observed coalition set $X = \{S_1, \ldots, S_n\}$ and center set $Z = \{T_1, \ldots, T_r\}$, write

$$\mathbf{K}_{ZZ} = \mathbf{L}\mathbf{L}^\top, \qquad \mathbf{F} = \mathbf{K}_{XZ}\mathbf{L}^{-\top}, \qquad \mathbf{Q} = \mathbf{F}\mathbf{F}^\top, \qquad \mathbf{D} = 10^{-6}\,\mathrm{diag}(N_1^{-1}, \ldots, N_n^{-1}).$$

The fit retains the fixed standardized mean zero, priors, optimizer, and noise, replacing $\mathbf{C}$ with $\mathbf{Q} + \mathbf{D}$ in the objective above. For N scheduled observations, including repetitions, we add the output-variance penalty

$$\frac{N\lambda s^2}{2}, \qquad \lambda = 0.01.$$

The penalty acts on the covariance output variance s. Let $k(S, Z)$ be the row of product-Hamming covariances, including output variance s. For fixed hyperparameters and centers, its feature coefficients are

$$\hat{\mathbf{w}} = (\mathbf{I} + \mathbf{F}^\top\mathbf{D}^{-1}\mathbf{F})^{-1}\mathbf{F}^\top\mathbf{D}^{-1}\mathbf{y},$$
$$\hat{f}(S) = \bar{y} + s_y k(S, Z)\mathbf{L}^{-\top}\hat{\mathbf{w}}.$$

We compute this fit and its objective by an augmented QR factorization, avoiding a full $n \times n$ training covariance. Predictions equal the fixed intercept $\bar{y}$ where correlations with all centers vanish. Although adding a constant to a fitted game leaves its marginal contributions unchanged, jointly estimating an intercept changes the kernel coefficients and can therefore change its Shapley variance.

Centers are selected by streamed pivoted Cholesky: at each step, choose the observed coalition maximizing the remaining diagonal $k(S, S) - k(S, Z)\mathbf{K}_{ZZ}^{-1}k(Z, S)$. Select centers under the prior-mode hyperparameters and retain them throughout optimization, coefficient fitting, and moment computation. After aggregating repeated observations, t evaluations of the objective and gradient cost $O(t(nr^2 + nrd + r^3))$ operations and $O(nr + nd + r^2)$ memory. Computing exact quadrature moments then costs $O(r^2d^2)$, with $O(d^3)$ node preparation, and predictions for correction cost $O(Nrd)$ for N scheduled rows.

Numerical moment computation. We use blocks of 128 kernel sections, extended-precision reductions, and the decomposition

$$\sigma_i^2(\hat{f}) = \mathbb{E}_t[\mathrm{Var}(\Delta_i\hat{f}(S) \mid t)] + \mathrm{Var}_t(\mathbb{E}[\Delta_i\hat{f}(S) \mid t]).$$

For section pairs, each coordinate's conditional covariance is $t(1 - t)[k_j(1, z_{aj}) - k_j(0, z_{aj})][k_j(1, z_{bj}) - k_j(0, z_{bj})]$. Prefix and suffix covariance recurrences avoid subtracting nearly equal full products; both variance terms use the same quadrature rule, preserving exactness of their combined integrand. A scaled roundoff estimate above 10^{-8} triggers extended-precision factors in place of the initial float64 factors. For finite-rank fits with $d \leq 16$, exceeding that threshold also triggers exhaustive extended-precision evaluation of the same fitted game, costing $O(2^d(rd + d^2))$ arithmetic operations and no game queries. Both computations record propagated roundoff estimates relative to $\max\{|\hat{\sigma}_i^2(\hat{f})|, s_y^2 s\}$, with scale one for constant responses. After precision fallback, these estimates are diagnostics rather than rejection criteria; they are floating-point error estimates, not interval-arithmetic certificates. Nonfinite results and negative intrinsic variances beyond roundoff tolerance remain failures; only negative intrinsic variances within tolerance are set to zero, while corrected estimates retain their sign. Prediction on N queried rows costs $O(Nrd)$, followed by $O(qd)$ for q chains' marginal contributions and correction, with no additional game queries.

B.3 Fractional Factorial Design

We generalize the aliasing method of Zhou et al. (2026) to estimate expectations under the Shapley distribution. Applying it to $\Delta_i f$ and $(\Delta_i f)^2$ yields a Shapley variance estimator with error controlled by higher-order Fourier terms. The estimator uses a fixed $O(d^2)$ query budget.

For $g : 2^{[n]} \to \mathbb{R}$, the target distribution chooses $|S|$ uniformly from $\{0, \dots, n\}$, then a uniform subset of that size. Let $g = \sum_T G(T)\chi_T$ be the Fourier decomposition of g.

Lemma B.2 (Fractional Factorial Exactness)**.** *For $n \geq 1$, Algorithm 3 uses $N = 2^{1+\lceil \log_2 n \rceil} < 4n$ evaluations and $O(n^2)$ additional arithmetic operations to return an output $\widehat{\mathbb{E}}[g]$ satisfying*

$$|\widehat{\mathbb{E}}[g] - \mathbb{E}[g(S)]| \leq \sum_{\substack{|T|\geq 4 \\ |T| \text{ even}}} |G(T)|.$$

The estimate is exact for every g of Fourier degree at most three, with error depending only on higher-degree terms. Figure 3 shows that higher-degree coefficients are often smaller, motivating the approximation. The bound depends on their total absolute magnitude, which need not be small even when individual coefficients are.

For $n = d - 1$, we use the same marginal queries for $g = \Delta_i f$ and $g = (\Delta_i f)^2$, then subtract the squared mean estimate from the second-moment estimate. This costs $2N < 8(d-1)$ game evaluations per player for $d \geq 2$.

Algorithm 3 gives the estimator; the aliasing identity below proves Lemma B.2.

Algorithm 3 Fractional factorial estimation of $\mathbb{E}[g(S)]$

Require: Oracle $g : 2^{[n]} \to \mathbb{R}$.
1: **if** $n = 0$ **then**
2: **return** $g(\emptyset)$.
3: $r \leftarrow \lceil \log_2 n \rceil$; $N \leftarrow 2^{r+1}$; $q \leftarrow 0$; $S \leftarrow [n]$.
4: Let $A_j \subseteq [r]$ index the nonzero bits of $j - 1$ for $j \in [n]$.
5: **for** each seed $z \in \{-1, +1\}^{r+1}$ in Gray-code order, starting with all signs $+1$ **do**
6: Update $S \leftarrow \{j : z_0 \prod_{k \in A_j} z_k = +1\}$ by flipping affected coordinates.
7: $q \leftarrow q + \left(\frac{2}{3N} + \frac{1}{6}\mathbf{1}[S \in \{\emptyset, [n]\}]\right) g(S)$.
8: **return** q.

With the labels of Algorithm 3, let $B(T) = \triangle_{j \in T} A_j$. The exact error is

$$\widehat{\mathbb{E}}[g] - \mathbb{E}[g(S)] = \sum_{\substack{|T|\geq 4 \\ |T| \text{ even}}} G(T) \left(\frac{1}{3} + \frac{2}{3}\mathbf{1}[B(T) = \emptyset] - \frac{1}{|T| + 1} \right). \tag{12}$$

For the size-uniform law on $[n]$, the independent-inclusion mixture from Appendix A.1 gives

$$\mathbb{E}[\chi_T(S)] = \frac{\mathbf{1}[|T| \text{ even}]}{|T| + 1}.$$

On the design, $\chi_T(S(z)) = z_0^{|T|} \chi_{B(T)}(z_1, \dots, z_r)$. Averaging over the seed signs gives zero for every odd-degree character; an even-degree character has design average $\mathbf{1}[B(T) = \varnothing]$ and endpoint average one. Its weight in the algorithm is therefore $\frac{1}{3} + \frac{2}{3}\mathbf{1}[B(T) = \varnothing]$. The constant has weight one, and distinct labels ensure $B(\{j, k\}) \neq \varnothing$, giving every pair weight $1/3$. Subtracting the target weights gives equation 12; every remaining weight difference has absolute value at most one. The empty and singleton labels recover every seed sign from its coalition, so the N queried coalitions are distinct and include both endpoints. Each Gray-code step updates at most n coalition signs, giving the arithmetic cost in Lemma B.2; accumulating values without storing past runs uses $O(n)$ working memory.

Fourier spectra. Figure 3 uses the estimation games, including neural-network Estate. For each player and function, normalize absolute Walsh coefficients by their maximum, including the constant, then sort by decreasing magnitude within each degree. Pad missing ranks with zeros and average over players and then thirty instances, retaining zero functions in the average. Equal-width degree blocks use a common logarithmic rank scale for both functions; exactly zero tails are omitted. For multiple backgrounds, square the marginal of the background-averaged game. Odd-degree terms integrate to zero for each plotted function, but the full marginal is squared before separating degrees: its odd terms can contribute even terms to the square.

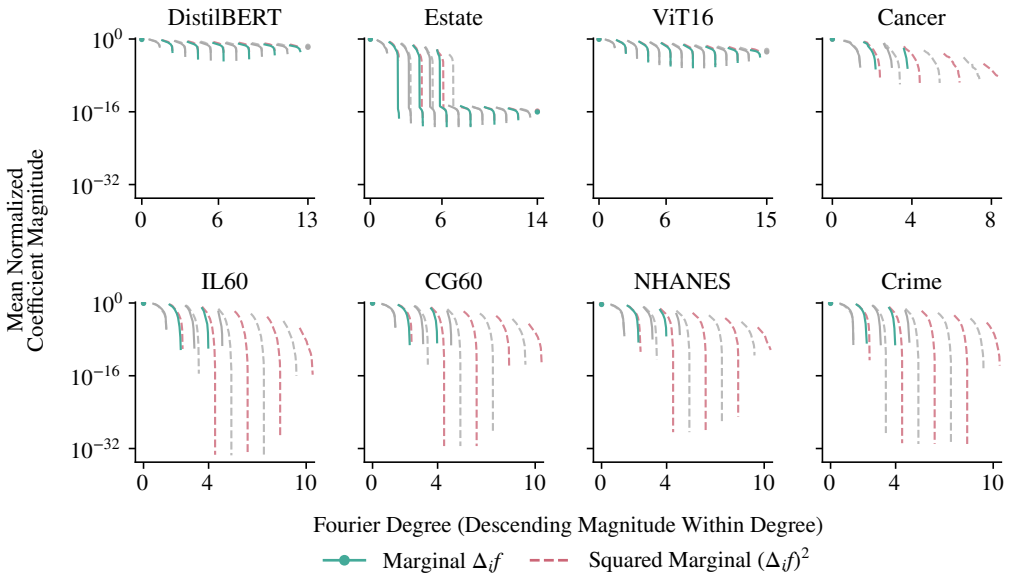

Figure 3: Fourier magnitudes ordered by degree, then decreasing magnitude within each degree. Gray odd degrees integrate to zero for each plotted function; FFD also integrates degrees zero and two exactly.

C Estimating Shapley Values

C.1 Size and Parity Decompositions

For $s \in \{0, \dots, d-1\}$, write the conditional marginal moments as

$$\mu_i(s) = \mathbb{E}[\Delta_i f(S) \mid |S| = s], \qquad \sigma_i(s) = \sqrt{\mathrm{Var}(\Delta_i f(S) \mid |S| = s)}.$$

Lemma C.1 (Size Stratification (Wu et al., 2023)). *For a fixed game f and player i, let $K = |S|$. Then*

$$\sigma_i^2(f) = \mathrm{Var}[\mu_i(K)] + \mathbb{E}[\sigma_i^2(K)]. \tag{13}$$

Proof. The identity is the law of total variance: conditioning on K makes $\Delta_i f - \mu_i(K)$ orthogonal to $\mu_i(K) - \phi_i(f)$. □

Lemma C.2 (Even and Odd Decomposition). *For a fixed game f and player i,*

$$\sigma_i^2(f) = \sigma_i^2(f_{\text{odd}}) + \sigma_i^2(f_{\text{even}}). \tag{14}$$

Proof. For the context complement $S^{c_i} = ([d] \setminus \{i\}) \setminus S$,

$$\Delta_i f_{\text{odd}}(S^{c_i}) = \Delta_i f_{\text{odd}}(S),$$
$$\Delta_i f_{\text{even}}(S^{c_i}) = -\Delta_i f_{\text{even}}(S).$$

The contexts S and S^{c_i} have equal Shapley probability. Pairing them in the expectation shows that the even marginal has mean zero and its product with the odd marginal has mean zero. Thus the two marginal components are uncorrelated, proving equation 14. □

C.2 Monte Carlo Shapley Error

Proof of Lemma 5.1. Both estimators are unbiased, so their vector error is the sum of their coordinate variances. Lemmas C.1 and C.2 give the unpaired identity. Each complementary pair averages to $\Delta_i f_{\text{odd}}(S)$, so $m/2$ independent bases give the paired identity.

Size stratification. Combining the within- and between-size decomposition of Wu et al. (2023, arXiv v5, Appendix B, Eqs. (B.6)–(B.8)) with parity gives four components. For $K = |S|$ and $p \in \{\text{even}, \text{odd}\}$, define

$$B_p = \sum_{i=1}^{d} \mathrm{Var}_K(\mathbb{E}[\Delta_i f_p(S) \mid K]),$$
$$W_p = \sum_{i=1}^{d} \mathbb{E}_K[\mathrm{Var}(\Delta_i f_p(S) \mid K)].$$

Then $\sum_i \sigma_i^2(f_p) = B_p + W_p$ by Lemma C.1. For unpaired stratification, allocate m/d independent uniform contexts to every size; for paired stratification, allocate $m/(2d)$ independent base contexts to every size and include their complements. Assume these counts are integers and draws are independent within and across sizes. With m/d independent samples per size, a coordinate's stratified variance is

$$\frac{1}{d^2} \sum_{s=0}^{d-1} \frac{\mathrm{Var}(\Delta_i f(S) \mid |S| = s)}{m/d} = \frac{1}{m} \mathbb{E}_K[\mathrm{Var}(\Delta_i f(S) \mid K)].$$

The conditional covariance of the even and odd marginals changes sign between sizes s and $d-1-s$, so its uniform average is zero. Summing over players gives $(W_{\text{even}} + W_{\text{odd}})/m$. Applying the same calculation to $m/2$ odd-marginal bases, balanced across sizes, gives $2W_{\text{odd}}/m$.

Permutation reuse. For a uniform permutation, the set of players preceding i is exactly $S \subseteq [d]\backslash\{i\}$ with probability $|S|!(d-1-|S|)!/d!$. Thus each player's marginal has the required Shapley law, and different permutations are independent. Sharing evaluations changes cross-player covariances, but $\mathbb{E}\|\boldsymbol{\phi} - \hat{\boldsymbol{\phi}}\|_2^2 = \sum_i \mathbb{E}[(\phi_i(f) - \hat{\phi}_i(f))^2]$ contains none of these covariances. For reverse-permutation pairs, independence is needed between base permutations; each pair's dependence is included in its averaged marginal.

C.3 Regression Identities

Regression fit. Let $d \geq 2$ and suppose $f(\emptyset)$ and $f([d])$ are known exactly. For each nonempty proper coalition S, let q_S be its probability on one draw, with $q_S = q_{S^c} > 0$ and $\sum_S q_S = 1$. Draw $n \geq d-1$ independent coalitions $S_1, \ldots, S_n$ from this distribution. Let $\boldsymbol{\beta} \in \mathbb{R}^d$ be the vector of fitted player coefficients and $\mathbf{1} \in \mathbb{R}^d$ the all-ones vector. With $w_S = [\binom{d}{|S|}|S|(d-|S|)]^{-1}$, the efficiency-constrained fit is

$$\hat{\boldsymbol{\phi}} = \underset{\mathbf{1}^\top\boldsymbol{\beta} = f([d]) - f(\emptyset)}{\arg\min} \sum_{j=1}^{n} \frac{w_{S_j}}{q_{S_j}} \left(f(S_j) - f(\emptyset) - \sum_{i \in S_j} \beta_i \right)^2 .$$

KernelSHAP uses $q_S \propto w_S$; leverage sampling uses $q_S = [(d-1)\binom{d}{|S|}]^{-1}$. The budget counts coalition rows with multiplicity; cached evaluations may cost less. The paired fit includes the complement of each of its $n/2$ independent base coalitions. Let $\mathbf{1}_S \in \mathbb{R}^d$ be the coalition indicator vector and set $\mathbf{a}_S = \mathbf{1}_S - (|S|/d)\mathbf{1}$. The sampled matrix is

$$\widehat{\mathbf{H}}_n = \frac{1}{n} \sum_{j=1}^{n} \frac{w_{S_j}}{q_{S_j}} \mathbf{a}_{S_j} \mathbf{a}_{S_j}^\top .$$

Its inverse acts on $\mathbf{1}^\perp = \{\mathbf{v} \in \mathbb{R}^d : \mathbf{1}^\top \mathbf{v} = 0\}$. Each expectation conditions on the sampled centered rows spanning this subspace; for paired regression, this is equivalent to the base rows spanning it. For $p \in \{\text{even}, \text{odd}\}$, define the sampled residual vector $\mathbf{r}_p \in \mathbb{R}^n$ and the columns of the regression map $\mathbf{W}_n \in \mathbb{R}^{d \times n}$ by

$$[\mathbf{r}_p]_j = f_p(S_j) - f_p(\emptyset) - \sum_{i \in S_j} \phi_i(f_p),$$

$$[\mathbf{W}_n]_{:,j} = \frac{w_{S_j}}{n q_{S_j}} \widehat{\mathbf{H}}_n^{-1} \mathbf{a}_{S_j}. \tag{15}$$

The map is the same for both parity games because it depends only on the sampled coalitions and weights. The normal equations give the following expanded expression for the unpaired regression risks defined in Section 5.2:

$$R_p(n) = \mathbb{E} \left\| \widehat{\mathbf{H}}_n^{-1} \frac{1}{n} \sum_{j=1}^{n} \frac{w_{S_j}}{q_{S_j}} \mathbf{a}_{S_j} \left[f_p(S_j) - f_p(\emptyset) - \sum_{i \in S_j} \phi_i(f_p) \right] \right\|_2^2 .$$

Population normal equations. Use the weights w_S and centered rows $\mathbf{a}_S$ defined above; all sums below run over nonempty proper coalitions. The full regression matrix and normal equations satisfy

$$\sum_S w_S \mathbf{a}_S \mathbf{a}_S^\top = \frac{1}{d}\left(\mathbf{I} - \frac{1}{d}\mathbf{1}\mathbf{1}^\top\right), \qquad \sum_S w_S \mathbf{a}_S \left(f(S) - f(\emptyset) - \sum_{i \in S} \phi_i(f) \right) = \mathbf{0}$$

(Musco & Witter, 2025, Lemma 3.3). The efficiency constraint places every fitted coefficient error in $\mathbf{1}^\perp$, where the full matrix is invertible.

Regression error decomposition. The residual used in $R_p(n)$ is $r_p(S) = f_p(S) - f_p(\emptyset) - \sum_{i \in S} \phi_i(f_p)$. Using $\boldsymbol{\phi}(f_{\text{even}}) = \mathbf{0}$ and $\boldsymbol{\phi}(f_{\text{odd}}) = \boldsymbol{\phi}(f)$ gives

$$r_{\text{even}}(S) = f_{\text{even}}(S) - \frac{f([d]) + f(\emptyset)}{2},$$

$$r_{\text{odd}}(S) = f_{\text{odd}}(S) - \sum_{i \in S} \phi_i(f) + \frac{f([d]) - f(\emptyset)}{2}.$$

The sampled normal equations give

$$\hat{\boldsymbol{\phi}} - \boldsymbol{\phi} = \widehat{\mathbf{H}}_n^{-1} \frac{1}{n} \sum_{j=1}^{n} \frac{w_{S_j}}{q_{S_j}} \mathbf{a}_{S_j} \big(r_{\text{even}}(S_j) + r_{\text{odd}}(S_j) \big).$$

Complementation negates $\mathbf{a}_S$ and $r_{\text{odd}}(S)$, while preserving $r_{\text{even}}(S)$ and $\mathbf{a}_S \mathbf{a}_S^\top$. Complementing all rows therefore negates the even error vector and preserves the odd error vector. The sampling law and full-rank event are complement invariant, so the expected inner product of these vectors is zero. Their squared norms have expectations $R_{\text{even}}(n)$ and $R_{\text{odd}}(n)$. For paired rows, the even sum vanishes and the sampled matrix equals the matrix from the $n/2$ base rows; the remaining squared error has expectation $R_{\text{odd}}(n/2)$. This proves Theorem 5.3. The separation under paired regression is also established by Fumagalli et al. (2026, Theorem 3.2).

Stratification counterexample. For $d = 3$, consider the game

$$f(S) = \begin{cases} 2, & S \in \big\{\{1\}, \{1,2\}, \{2,3\}, [3]\big\}, \\ 0, & \text{otherwise}, \end{cases} \qquad \boldsymbol{\phi} = (1, 1, 0).$$

With a budget of two queries, compare uniform sampling with one uniform singleton and one uniform pair, rejecting rank-deficient designs before evaluating f. Their errors are

$$\mathbb{E}\|\boldsymbol{\phi} - \hat{\boldsymbol{\phi}}^{\text{unpaired}}\|_2^2 = \frac{8}{3} < \frac{10}{3} = \mathbb{E}\|\boldsymbol{\phi} - \hat{\boldsymbol{\phi}}^{\text{stratified}}\|_2^2.$$

Both fits are unbiased; stratification raises variance by 25%.

To compute these errors, subtract $\mathbf{1}[1 \in S] + \mathbf{1}[2 \in S]$ from the game. The residual is even with zero endpoints and values $h_1 = 1$, $h_2 = -1$, $h_3 = 0$ on singleton i and its complement, so the original Shapley vector is $(1, 1, 0)$. Write $\mathbf{a}_i = \mathbf{e}_i - \mathbf{1}/3$. A singleton row has direction $\mathbf{a}_i$ and its complement has direction $-\mathbf{a}_i$. The two-row design has full rank exactly when its indices i, j differ, an event of probability $2/3$ under either sampling law. Rejecting deficient designs before querying the game keeps the cost at two coalition values. For a full-rank design, let $\varepsilon_i, \varepsilon_j \in \{-1, 1\}$ mark the row directions. Interpolation and efficiency give error coordinates $\varepsilon_i h_i$, $\varepsilon_j h_j$, and $-\varepsilon_i h_i - \varepsilon_j h_j$, hence

$$\|\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}\|_2^2 = 2(h_i^2 + h_j^2) + 2\varepsilon_i \varepsilon_j h_i h_j.$$

For uniform distinct indices, $\mathbb{E}[h_i^2 + h_j^2] = 4/3$ and $\mathbb{E}[h_i h_j] = -1/3$. Unpaired sampling has independent uniform row signs, giving error $8/3$. With one singleton and one pair, $\varepsilon_i \varepsilon_j = -1$, giving error $8/3 + 2/3 = 10/3$. Complementing both rows negates the residual fit while preserving either sampling law and the full-rank event, so both estimators are unbiased.

D Experimental Setup

D.1 Games and Exact References

Table 2 lists the eight descriptive games, with thirty explained instances per family. The estimation benchmarks retain seven of these models and replace Estate's boosted trees with a neural network having two hidden layers of 64 units and standardized inputs and targets. Tabular predictors use the `shapiq` train/test split and a fixed background of fifty training rows:

$$f(S) = \frac{1}{50} \sum_{b=1}^{50} h(x_S, x^{(b)}_{[d]\setminus S}).$$

The explained row x belongs to the test split. Tree predictors use XGBoost library defaults and raw model outputs; absent transformer tokens or patches are masked. The three games with $d \leq 16$ use exhaustive coalition enumeration for ground truth.

Game	Model and value function	d	Ground truth	Fourier terms
DistilBERT	DistilBERT (IMDB), signed sentiment score, tokens masked	14	enumeration	2^{14}
Estate	XGBoost regressor, interventional	15	enumeration, readout	279
ViT16	ViT-B/32-384, class probability, 4×4 patches masked	16	enumeration	2^{16}
Cancer	XGBoost classifier (logit), interventional	30	readout	286
IL60	XGBoost regressor, interventional	60	readout	7,091
CG60	XGBoost regressor, interventional	60	readout	9,334
NHANES	XGBoost regressor, interventional	79	readout	2,805
Crime	XGBoost regressor, interventional	101	readout	14,637

Table 2: Descriptive games and median nonzero Fourier-coefficient counts over thirty instances. The estimation benchmarks replace Estate's trees with a neural network; readout denotes exact Fourier computation.

Tree ground truth. We merge Fourier coefficients across leaves, trees, and background rows before applying Lemma 3.1; averaging background-specific variances would omit covariance between backgrounds. Enumeration of Estate's Fourier representation agrees with its analytic variance to $2 \cdot 10^{-14}$, scaling maximum coordinate error by the largest reference variance. Live float32 predictions differ slightly because of accumulation and tree-parser rounding; the checked Estate instance has normalized squared variance error $2.3 \cdot 10^{-11}$. Validation scripts and numerical diagnostics are included in the repository.

D.2 Application Games

Local Explanations. The local-explanation comparison retains sixteen explicands from each original family and adds fourteen tabular datasets with TabPFN-3 (Grinsztajn et al., 2026), random forests, ExtraTrees, and multilayer perceptrons. Players are input features, tokens, or image patches. The cached TabPFN games predict a class-one logit or regression output with absent features set to positive infinity; the other new tabular models use training-median replacement and class-one probability or regression output. The original families retain their existing game definitions. This expanded population differs from the original 160-game study in Table 3.

Feature Retraining. Feature retraining uses sixteen datasets with linear/logistic models, LightGBM, random forests, and TabPFN-3, except that ExtraTrees replaces the coalition-constant Amazon linear setting. Players select columns used to fit the predictor, with source-defined baselines for empty or degenerate selections. Sixteen fixed validation batches rescore each setting's coalition-specific predictors. The utility is validation accuracy or negative mean squared error on datasets from TabArena (Erickson et al., 2025), with the separate sklearn diabetes regression dataset also included.

Training-Group Valuation. Training-group valuation has 45 TabPFN-3 and nineteen linear/logistic settings on TabArena datasets; players select groups of training rows. Coalition-specific predictors are scored by validation accuracy or negative mean squared error on sixteen fixed validation batches, retaining source-defined baselines for empty or degenerate selections.

Ensemble Selection. Ensemble selection has 51 LIGHTGBM and thirteen linear/logistic settings on TabArena datasets, each with sixteen pretrained bootstrap members as players. Coalitions average member probabilities before classification, or average regression predictions, and receive validation accuracy or negative mean squared error.

Transformer Ablations. Transformer games use 64 released semi-synthetic InterpBench task/model settings (Gupta et al., 2024), each evaluated on sixteen independently seeded input batches. Players are attention heads and MLP blocks; excluded components receive their clean-batch mean activations, and utilities average target log-probability for categorical outputs or negative squared error for numerical outputs over defined task outputs.

Population. Each construction contains 64 distinct dataset/model or task/model settings with sixteen instances per setting, giving 5,120 games across the five constructions. Instances are explained rows, fixed validation batches, or sampled transformer input batches; estimator repetitions are a separate source of randomness. Tabular instances share fitted models and training data. Initial validation batches are disjoint where possible; smaller validation sets and reserve replacements use fixed subsets sampled without replacement within each batch, with overlap allowed across batches. These instances are not independent datasets or independently trained models.

We restrict the final comparison to games whose exact Shapley-variance vector has positive squared norm, so normalized error is defined. A frozen, truth-only reserve rule replaced 93 exactly zero-variance tabular games: thirty feature-retraining, seventeen training-group, and 46 ensemble games. The rule retains every positive-variance game and first tries distinct validation batches from a fixed seed stream; settings with coalition-constant per-row utilities are replaced from a prescribed dataset/model reserve list. Acceptance uses exact zero rather than a tolerance, and never uses estimator errors. The resulting population still has 64 settings and sixteen instances per construction; the rejected games and all reserve attempts remain archived.

Exact references. A checksum-pinned catalog records each game's player order, utility, data, model, and construction protocol. The TabPFN explanation corpus pins its own inputs and predictors. Archived tabular prediction/loss components define fixed utilities; they do not establish that the original training environments can be recovered or that validation rows are paired across models. Complete coalition tables give exact references by summing weighted centered squared marginal contributions; the original high-dimensional tree games retain their exact Fourier references. The complete-table games test transfer across constructions at player counts that permit enumeration; they do not establish scalability for large non-tree games. Estimators receive only the value oracle, without unqueried table entries or reference moments.

D.3 BUDGETS AND EVALUATION

Variance error. We report $\|\boldsymbol{\sigma}^2(f) - \hat{\boldsymbol{\sigma}}^2(f)\|_2^2 / \|\boldsymbol{\sigma}^2(f)\|_2^2$. The original eight-family study averages three repetitions within each of twenty evaluation instances (indices 10–29) per family; deterministic FFD uses one estimate. Its ablation table reports medians across instance means, and Table 3 gives medians and interquartile ranges. Original-study quartiles use linear interpolation between ordered instance errors. The benchmark uses separately seeded MC runs; configuration selection uses MC on the same chains. Evaluation instances are excluded from parameter selection; earlier evaluation results informed estimator revisions, so they are not untouched data for method development.

Application aggregation. For the 5,120-game comparison, we average normalized squared error over three repetitions within each game; deterministic FFD uses one estimate. Datasets receive equal weight within each application, model/protocol settings receive equal weight within each dataset, and the sixteen instances receive equal weight within each setting. Overall summaries give equal weight to the five applications. Table 1 gives the weighted 25th percentile, median, and 75th percentile of nMSE. Quartiles use the inverse empirical cumulative distribution without interpolation and describe

spread across games rather than confidence intervals. An aggregate for a method and application requires every scheduled game and repetition. Every measured error, including outliers, remains in the summaries. Medals compare unrounded nMSE independently within each statistic and application, with ties sharing a medal; the original-family table compares medians. For the game-wise ratios used in the diagnostic discussion, we divide each game's mean error by its corresponding MC average before aggregation. A statistic of these ratios need not equal the ratio of the corresponding nMSE statistics, and a zero MC error leaves the ratio undefined without a denominator floor.

Query accounting. Tables 1 and 3 use the full FFD cost $B_{\mathrm{FFD}}(d) = 2Nd$, where $N = 2^{1+\lceil \log_2(d-1) \rceil}$. For the original families, these budgets are $64d$ for DistilBERT, Estate, and ViT; $128d$ for Cancer; $256d$ for IL60 and CG60; and $512d$ for NHANES and Crime. The stochastic estimators also use $4d$ and $16d$ allowances in Figures 4 and 5; FFD uses its full-design budget only. Budgets count endpoints and repeated sampled coalitions; memoization saves evaluations without supplying extra draws. MC and the three surrogate estimators evaluate endpoints once, then use $d-1$ interior queries per permutation, leaving fewer than $d-1$ unused queries. The corrected surrogate estimators use the same query allowances as MC.

Computation time. Figure 5 measures warmed estimator computation and cached-query time on 4,992 complete-table games: 896 local explanations and 1,024 games in each other application. The 128 original-roster instances remain in the query and Table 1 comparisons but are excluded from this figure because their earlier timings follow a different protocol. Error quantiles and mean time coordinates are recomputed with the application weights on this fixed population. Imported timings use audited replays of the same estimates and queries; fresh runs preload estimator dependencies. Library initialization and original model inference or retraining are excluded. The points retain their fixed query allowances and do not compare methods after equal elapsed time.

D.4 Surrogate Settings

Trees and polynomials. σ^2SHAP [Tree] uses 500 depth-three trees at learning rate 0.1 at every budget. We select it against the transferred budget-dependent setting and 1,000 depth-six trees at learning rate 0.1, using instances 2 and 3 of all eight families at their FFD budgets, with three repetitions and matched observations and seeds. Among configurations whose mean and quartiles of instance-wise error ratios to same-chain MC are below one, we choose the lowest mean; the configuration is frozen before validation on instances 4 and 5. Those validation instances were reserved for this selection decision, but earlier GP studies used them. σ^2SHAP [Polynomial] uses 1,000 depth-five trees at learning rate 0.03 for $m \in \{4, 16, 64, 128\}$ and 500 depth-six trees at learning rate 0.1 for $m \in \{256, 512\}$. These settings transfer from development comparisons of earlier surrogate variants at $m = 64$ and $m = 512$, using instances 0–9 and choosing by mean within-instance error rank across families; each setting transfers to the nearest budget on a logarithmic scale. The candidate grid crossed depths $3, 5, 6, 8$ with tree-count/rate pairs $(100, 0.3)$, $(300, 0.1)$, $(1000, 0.03)$, and $(500, 0.1)$, plus four regularized settings with depth six or eight, 300 or 500 trees, rate 0.1, minimum child weight five, and ridge regularization five.

Gaussian processes. σ^2SHAP [GP] uses the dense fit through $16d$ scheduled observation rows and the regularized finite-rank fit above that threshold, with penalty $\lambda = 0.01$ and at most 128 centers. Penalty selection used instances 0 and 1 of all families at $4d$, $16d$, and $64d$, comparing zero penalty and $\lambda_0(d/32)^p$ for $\lambda_0 \in \{10^{-4}, 10^{-2}, 1, 100\}$ and $p \in \{-1, 0, 1\}$, preferring a shared strength within 10% of the best score. Center caps 128, 256, and 512 were compared on DistilBERT, Cancer, CG60, and Crime instances 7 and 8 at $64d$ and $256d$, minimizing the geometric mean cell error ratio to same-chain MC among caps with complete computations. The selected cap was validated on the other four families at those indices and budgets; these families were held out from center-count selection only. The penalty and cap transfer from an earlier fitting variant; the current fit retains its optimization centers, as specified in Appendix B.2.3. We use exact variance to score development configurations, but not in fitting or deployment.

Coverage and fit versions. The initial five-application campaign produced all 66,560 FFD-budget estimates and 122,880 smaller-budget estimates, with no timeouts. Table 1 replaces all 15,360 GP FFD-budget estimates with matched fixed-mean refits, also without failures or timeouts. The original

eight-family table and all GP points in Figures 4 and 5 retain the previous free-mean fit while their matched reruns are pending. The application comparison retains the frozen MC, FFD, Tree, and Polynomial estimators from the original study. GP retains its kernel, priors, center rule, and penalty; the fixed-mean fit sets its intercept to the sample mean during fitting. Surrogate corrections reuse fitting queries and do not guarantee lower estimation error. Each initial application attempt has a 1,800-second wall-clock limit, including worker startup and game loading; matched GP refits apply this limit to estimator computation after loading the archived observations. Isolated process failures were retried with the same samples and settings; missing, failed, and timed-out attempts remain archived. Independent audits verify the correspondence among source identities, query transcripts, and exact references, and recompute normalized errors and aggregated ratios from the saved estimates.

The earlier eight-family GP campaign produced 2,427 successful estimates, three numerical failures, and thirty timeouts from 2,460 planned cases under a 30-minute estimator-computation limit. All eight FFD-budget cells in Table 3 are complete; incomplete smaller-budget cells remain archived. Numerically accepted fits are retained regardless of optimizer convergence. The acceptance threshold and extended-precision fallback are specified in Appendix B.2.3.

D.5 SHAPLEY VALUE EVALUATION

We use native `shapiq` implementations with local adapters for reversed permutations and FFD. For each of 240 instances, methods are ranked by mean squared vector error over the first twenty saved repetitions, with midranks for ties; FFD uses one deterministic run. Ranks are averaged within families and then equally across eight families. No method receives external hyperparameter search; native support selection and internal cross-validation remain part of each estimator. RegressionMSR and ProxySPEX use fixed XGBOOST proxies of 100 depth-six trees at learning rate 0.3. OddSHAP retains its native LIGHTGBM configuration: 100 trees, maximum depth ten, 31 leaves, and learning rate 0.1.

Pairing comparison. In Figure 2, colors distinguish game families; each arm uses $64d$ queries and 100 repetitions per instance. The fit is linear in logarithms, and its bands are computed by resampling whole families. Held-out R^2 pools predictions made by withholding each family in turn and is measured on the logarithmic response scale. Native LeverageSHAP samples without replacement in its unpaired arm; native RegressionMSR uses its in-sample proxy correction. These protocols differ from the independent-row identities in Theorem 5.3.

E Additional Results

E.1 Original Local-Explanation Comparison

Method Budget	DistilBERT ($\times 10^{-1}$) $64d$	Estate ($\times 10^{-2}$) $64d$	ViT16 ($\times 10^{-1}$) $64d$	Cancer ($\times 10^{-2}$) $128d$	IL60 ($\times 10^{-3}$) $256d$	CG60 ($\times 10^{-3}$) $256d$	NHANES ($\times 10^{-3}$) $512d$	Crime ($\times 10^{-3}$) $512d$	% of MC All games
σ^2SHAP [Polynomial]	.85 (.44–1.7)	.069 (.040–.14)	.30 (.21–.48)	.062 (.044–.15)	.59 (.48–.92)	1.8 (1.6–2.7)	.37 (.20–.66)	1.1 (.67–1.3)	**22%** (**8.7–51**)
σ^2SHAP [GP]	.80 (.49–1.8)	.029 (.017–.074)	.29 (.22–.45)	.084 (.060–.17)	1.2 (.91–1.7)	3.3 (2.5–4.7)	1.2 (.63–1.6)	2.0 (1.2–2.7)	**38%** (**14–75**)
σ^2SHAP [Tree]	.87 (.48–1.7)	.33 (.20–.66)	.35 (.24–.62)	.58 (.35–.73)	1.9 (1.4–2.4)	3.4 (2.7–5.0)	1.3 (.91–1.7)	1.5 (.93–3.4)	**55%** (**28–93**)
FFD	1.5 (.62–3.0)	.35 (.053–.61)	.97 (.47–2.0)	.033 (.014–.054)	2.9 (1.6–7.4)	6.2 (4.6–8.7)	2.8 (1.5–4.8)	2.2 (1.1–4.4)	**90%** (**25–160**)
MC	.88 (.61–1.6)	1.9 (.98–2.9)	.38 (.27–.67)	.86 (.72–1.3)	5.3 (3.9–7.3)	5.3 (4.4–6.6)	2.9 (2.4–3.8)	2.6 (2.0–4.4)	**100%** (**100–100**)

Table 3: Original local-explanation study: median nMSE and 25th–75th percentiles across twenty instances per family at full FFD budgets. Column multipliers apply to both nMSE lines; the final column summarizes instance-wise percentages of MC error across all 160 games. Medals compare unrounded medians. GP entries retain the previous fit while the fixed-mean rerun is pending.

E.2 Query and Computation Budgets

Figures 4 and 5 give one panel per application, using the populations and weights in Appendix D.3. All GP curve points use the previous free-mean fit; Table 1 uses the matched fixed-mean refits at full FFD budgets. The median curves and interquartile bands retain every game's error but do not display the extreme GP tails examined in Appendix E.4.

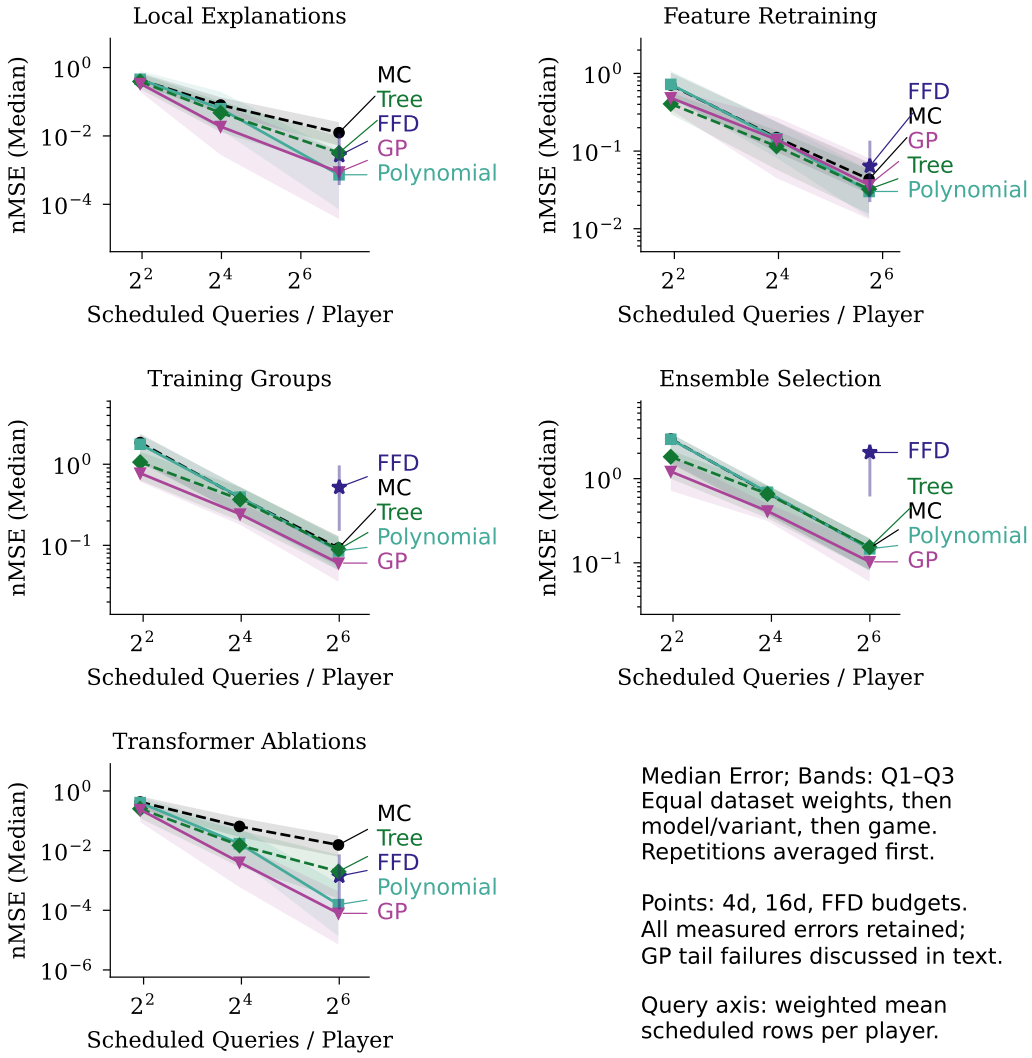

Figure 4: Median nMSE against scheduled queries per player on all 5,120 games, with one panel per application; bands show the 25th–75th percentiles. Points use $4d$, $16d$, and full FFD allowances, with one FFD point per application. Error quantiles and mean query coordinates use the dataset/model/instance weights in Appendix D.3; the FFD query coordinate varies with player count. Tree, Polynomial, and GP abbreviate the corresponding σ^2SHAP estimators, with the previous free-mean fit used for GP. The relative benefit of each surrogate depends on the application and query allowance.

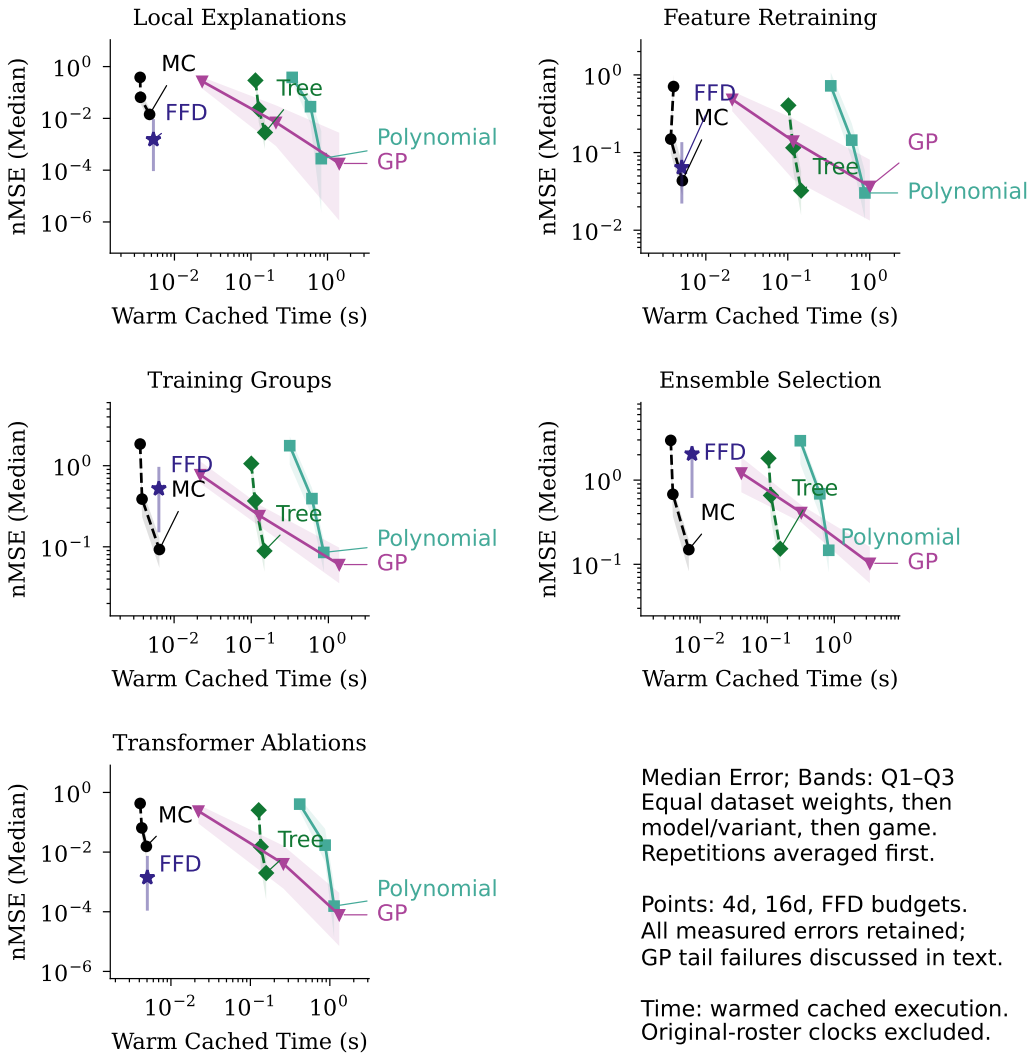

Figure 5: Median nMSE against warmed estimator computation and cached-query time, with one panel per application; bands show the 25th–75th percentiles. The fixed population has 4,992 complete-table games: 896 local explanations and 1,024 games in each other application. The 128 original-roster instances are excluded because their earlier timings follow a different protocol. Error quantiles and mean time coordinates use the application weights on this population. Timings exclude library initialization and original model inference or retraining; points retain their query allowances and do not compare methods after equal elapsed time. GP uses the previous free-mean fit. Additional surrogate computation reduces typical error on some applications.

E.3 Higher-Order Structure and Transfer

Tree and Polynomial settings were selected on local-explanation games and transferred unchanged. FFD integrates odd Fourier terms and degrees zero and two exactly; higher even-degree terms in marginal contributions and their squares determine the remaining moment errors. We examine one preselected instance from each complete-table setting: 56 additional tabular local-explanation settings and 64 settings from each other application, totaling 312 games. For each game, we sum squared Fourier coefficients of squared marginal contributions over players and measure the share at even degrees at least four relative to all nonconstant degrees. The median share is 0.19% for local explanations and 0.37% for transformer ablations, versus 39.7% for training-group valuation and 46.8% for ensemble selection. Across the matched games, this share has Spearman correlation

0.885 with FFD's error ratio to MC. These pilot statistics give settings equal weight, separately from the weighted full-benchmark comparison. They support an association between FFD error and higher-order structure in squared marginals; coefficient signs and aliasing patterns also matter, so high-order energy alone does not determine the error.

The empty-predecessor stratum contributes a median 65.6% and 86.1% of total player variance in the training-group and ensemble pilots. FFD overweights this context: at $d = 12$ and $d = 16$, its empty-context weight is $3/16$, compared with the Shapley probability $1/d$. For the illustrative game $f(S) = \mathbf{1}[S \neq \emptyset]$, this produces variance estimates 1.99 and 2.60 times the truth, respectively. This overweighting of the empty context is consistent with the measured losses; it does not isolate the cause of error for every benchmark game.

E.4 GP Failure Analysis

Median and interquartile summaries describe typical performance; extreme errors remain in the evaluation. The previous free-mean GP had an overall mean game-wise error ratio of $6.6 \times 10^6\%$ of MC; feature retraining contributed the largest failures, with a mean ratio of $3.3 \times 10^7\%$, while transformer ablations had a mean ratio of $7.4 \times 10^3\%$. For that free-mean fit, ten feature-retraining games account for 98.1% of that construction's summed normalized error. In the worst repeated fit, true coalition accuracies lie between 0.9375 and 0.96875, while the fitted surrogate reaches 3.07 on unobserved coalitions. Independent full-coalition calculations reproduce the fitted variances in four checked fits; a sixty-digit coefficient solve also reproduces the worst fit. The reused residual samples can miss these off-sample errors, so a small training residual does not ensure accurate variance estimation. Short learned lengthscales left some unobserved coalition patterns weakly correlated with every center. The free intercept and kernel terms cancelled on observed patterns, while predictions on missing patterns approached the large intercept. Fixing the intercept during fitting changes the compensating kernel coefficients; shifting the intercept after fitting would leave every marginal contribution unchanged. On three selected tail traces, fixed-mean hyperparameter refitting reduced nMSE from millions to 0.252, 0.0163, and 0.126, compared with same-trace MC errors of 0.275, 0.0489, and 0.166. Independent sixty-digit coefficient solves reproduced these results; three fixed controls showed mixed small changes. These six diagnostic traces motivated the revised fit and do not establish benchmark-wide improvement. The matched rerun over all 5,120 games gives GP an overall mean error ratio of 83.5% of MC and a median of 47.3%, with 25th–75th percentiles of 3.4%–94.0%. For feature retraining, the mean ratio remains 200.2% and the 75th percentile is 123.8%. All game errors are retained in these summaries.

E.5 Shapley Value Estimator Comparison

The five leading methods, OddSHAP, LeverageSHAP, FFD, RegressionMSR, and Antithetic Permutation, all use complementary samples. This ranking alone does not isolate the benefit of pairing: OddSHAP and FFD have no native unpaired control here. The matched comparison in Figure 2 covers MC, LeverageSHAP, and RegressionMSR; Appendix D.5 gives both protocols.

Name	Citation	Surrogate	Pairing	Note	Average Rank
OddSHAP	(Fumagalli et al., 2026)	Odd Polynomial	✓	Paired Sparse Fourier Regression	1.62
Leverage SHAP	(Musco & Witter, 2025)	Linear	✓	Leverage-Score Sampling	3.38
FFD	(Zhou et al., 2026)	—	✓	Intrinsic Pairing; Possible Aliasing	3.58
RegressionMSR	(Witter et al., 2025)	Trees	✓	Native In-Sample Correction	3.67
Antithetic Permutation	(Mitchell et al., 2022)	—	✓	Reversed Permutation Pairs	5.19
KernelSHAP	(Lundberg & Lee, 2017)	Linear	×	Kernel-Weighted Regression	6.61
MC	(Castro et al., 2009)	—	×	Shared Permutation Chains	7.00
Stratified Sampling	(Castro et al., 2017)	—	×	Coalition-Size Strata	8.58
SPEX	(Kang et al., 2025)	Sparse Fourier	×	Structured Fourier Recovery	9.01
Stratified SVARM	(Kolpaczki et al., 2024)	—	×	Native `SVARM` Class	9.37
Owen Sampling	(Okhrati & Lipani, 2021)	—	×	Quadrature Over Inclusion Probabilities	9.40
ProxySPEX	(Butler et al., 2025)	Sparse Fourier	×	Fixed Tree Configuration; Native Refit	11.79
Unbiased KernelSHAP	(Covert & Lee, 2021)	Linear	×	Known Population Gram Matrix	11.80

Table 4: Thirteen Shapley value estimators on 240 instances from the original eight local-explanation families, at each game's full FFD budget. Surrogates are fitted by the estimators; pairing includes complementary coalitions, reversed permutations, and FFD's intrinsic design.

E.6 VARIANCE ESTIMATOR ABLATIONS

Table 5 compares matched variants on the original eight local-explanation families at their full FFD budgets. Uncorrected rows return the exact variance of the corresponding corrected estimator's fitted surrogate on the same permutation chains. Correction lowers all eight family medians for both the tree and final polynomial surrogates.

The corrected Fourier-only variant stops after the initial Fourier least-squares fit; adding the final Möbius refit lowers median error in seven families, with Crime the exception. The no-Fourier-refit variant preserves the initial support and its query-row and rank filters, but converts the original tree coefficients directly before the final Möbius fit and correction. Omitting this fit increases Estate's median error by a factor of 1.987, with five families tied to numerical precision.

The tree-selection variant expands all tree leaves directly into Möbius monomials, aggregates their coefficients, and selects up to 256 nonconstant terms by absolute coefficient magnitude. It uses the same final row-count and rank safeguards, Möbius fit, exact variance computation, and correction. All eight family medians worsen relative to σ^2SHAP [Polynomial]. These ablations share query transcripts, tree seeds, and frozen settings; improvements at FFD budgets need not hold at every smaller budget.

Method	DistilBERT	Estate	ViT16	Cancer	IL60	CG60	NHANES	Crime
Query Budget	$64d$	$64d$	$64d$	$128d$	$256d$	$256d$	$512d$	$512d$
MC	8.8×10^{-2}	1.9×10^{-2}	3.8×10^{-2}	8.6×10^{-3}	5.3×10^{-3}	5.3×10^{-3}	2.9×10^{-3}	2.6×10^{-3}
σ^2SHAP [Tree]	8.7×10^{-2}	3.3×10^{-3}	3.5×10^{-2}	5.8×10^{-3}	1.9×10^{-3}	3.4×10^{-3}	1.3×10^{-3}	1.5×10^{-3}
Tree (chains, uncorrected)	4.0×10^{-1}	1.2×10^{-2}	2.1×10^{-1}	2.8×10^{-2}	4.2×10^{-2}	1.2×10^{-1}	9.5×10^{-2}	1.7×10^{-1}
σ^2SHAP [Polynomial]	8.5×10^{-2}	6.9×10^{-4}	3.0×10^{-2}	6.2×10^{-4}	5.9×10^{-4}	1.8×10^{-3}	3.7×10^{-4}	1.1×10^{-3}
Möbius (chains, uncorrected)	2.4×10^{-1}	1.6×10^{-3}	6.4×10^{-2}	1.4×10^{-3}	2.0×10^{-3}	1.9×10^{-2}	4.5×10^{-3}	2.1×10^{-2}
σ^2SHAP [Polynomial] (Fourier refit)	8.8×10^{-2}	1.0×10^{-3}	3.2×10^{-2}	1.1×10^{-3}	6.7×10^{-4}	1.9×10^{-3}	3.9×10^{-4}	1.1×10^{-3}
Möbius (no Fourier refit)	8.6×10^{-2}	1.4×10^{-3}	3.0×10^{-2}	6.2×10^{-4}	5.9×10^{-4}	1.8×10^{-3}	3.7×10^{-4}	1.1×10^{-3}
Möbius (tree selection)	8.9×10^{-2}	2.1×10^{-3}	3.2×10^{-2}	2.1×10^{-3}	1.5×10^{-3}	6.7×10^{-3}	9.4×10^{-4}	1.6×10^{-3}

Table 5: Matched ablations at full FFD budgets: median instance-mean normalized squared variance error. Uncorrected rows use the same fitted surrogates as their corrected counterparts. The remaining variants isolate the polynomial refits or replace Fourier support selection by direct tree-to-Möbius selection.

F Explanation Examples

F.1 Figure 1 Protocol

Housing. A random forest predicts individual sale prices in Ames Housing (De Cock, 2011) from above-ground living area, age at sale, total above-ground rooms, and above-ground bedrooms; room counts exclude bathrooms. It uses 100 trees, maximum depth eight, minimum leaf size five, and seed 42 in scikit-learn 1.9.0. We retain detached single-family houses with nonnegative age and complete inputs, splitting 1,939 training and 485 held-out houses with seed 42. Coalition values average over 128 fixed training backgrounds; all sixteen coalitions are evaluated exactly. Figure 1 explains source row 948 (Order 949): a five-year-old house with 1,710 square feet, seven rooms, and three bedrooms. Its sale price is \$240k; the model predicts \$216k. We selected it from held-out houses for similar mean contributions of rooms and bedrooms and different variances, with the model and background fixed. The selected means differ by 9%, while the bedroom variance is 7.66 times larger.

Image. We crop the author's photograph of Ivy to a square containing her face and front paws. ViT-B/16-384 predicts the probability of its unmasked-image class, Egyptian cat, replacing absent patch embeddings by zero. The 6×6 players group its 24×24 patch tokens into four-by-four blocks. A floor patch and a front-paw patch were selected for similar means and different distributions using 1,024 preliminary permutations. Displayed moments and histograms use 2,048 fresh independent uniform permutations; bins retain every sampled contribution, with independent probability scales.

Transparency. Let $u_i = \sigma_i^2(f)/(\phi_i(f)^2 + 10^{-8})$ in native output units: probabilities for the image and units of 100,000 dollars for housing. The image opacity multiplier is $(1 + \min_j u_j)/(1 + u_i)$, applied to a fixed overlay opacity of 0.72. For housing, set $\ell_i = \log(u_i + 10^{-8})$ and use $0.2 + 0.8(\max_j \ell_j - \ell_i)/(\max_j \ell_j - \min_j \ell_j)$, or unit opacity when the denominator vanishes. Attributions with lower relative variance are more opaque; signed colors and attribution magnitudes remain unchanged. These selected examples illustrate context variation, not estimation uncertainty, and do not establish improved explanatory faithfulness.

F.2 Text Explanation

Figure 6 extends the display to instance zero of the descriptive DistilBERT game, using exact marginal distributions and moments. Player A has the largest absolute Shapley value; player B has the largest $\sigma_i(f)/|\phi_i(f)|$ among players with $|\phi_i(f)| > 0.05 \max_j |\phi_j(f)|$. Both attribution panels share signed colors and magnitudes. The right panel multiplies base opacity by $0.2 + 0.8 q_i / \max_j q_j$, where $q_i = \phi_i(f)^2/(\phi_i(f)^2 + \sigma_i^2(f))$. We set $q_i = 1$ when both moments vanish and use multiplier 0.2 if every q_i is zero.

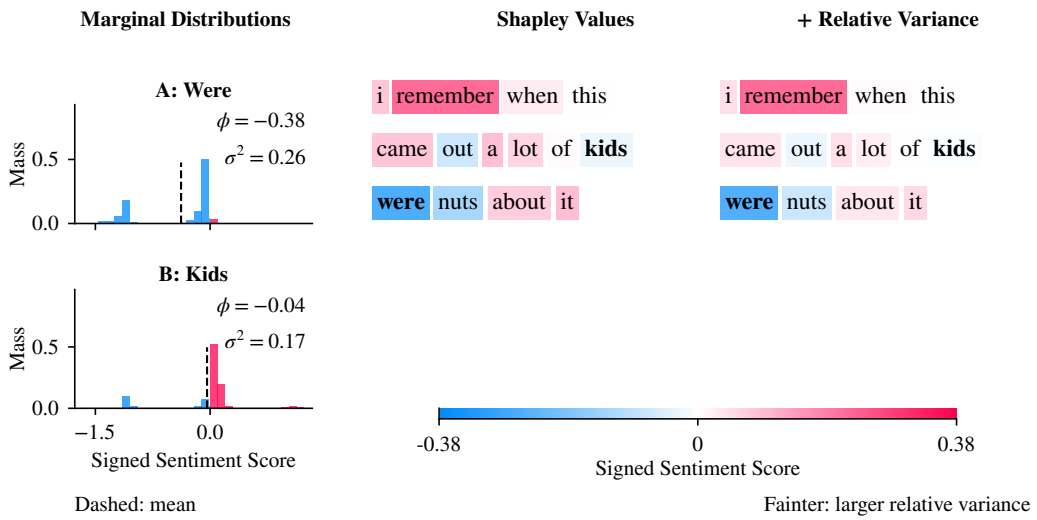

Figure 6: DistilBERT explanation with exact marginal distributions for “were” (A) and “kids” (B). The output is a signed sentiment score; the right panel attenuates contributions with higher variance relative to their squared means.

G Limits of Computation

G.1 Value-Oracle Approximation Lower Bound

An unqueried coalition can carry an arbitrarily large value, so sampling alone cannot give a uniform accuracy guarantee over unrestricted games.

Lemma G.1 (Approximation Requires Exponentially Many Queries). *Fix $d \geq 2$, a player $i \in [d]$, and either an additive tolerance $\epsilon \geq 0$ or a multiplicative factor $C \geq 1$, both finite. Suppose a randomized value-oracle algorithm returns $\hat{\sigma}_i^2(f)$ and, for every game $f : 2^{[d]} \to \mathbb{R}$, satisfies with probability at least $2/3$ either*

$$|\hat{\sigma}_i^2(f) - \sigma_i^2(f)| \leq \epsilon, \qquad \text{or} \qquad C^{-1}\sigma_i^2(f) \leq \hat{\sigma}_i^2(f) \leq C\sigma_i^2(f),$$

respectively. Even when $f(\emptyset)$ and $f([d])$ are supplied for free, the algorithm makes at least $(2^d-2)/3$ queries in expectation on the zero game. Consequently, any worst-case query bound is $\Omega(2^d)$.

Proof. Let $\mathcal{T} = 2^{[d]} \setminus \{\emptyset, [d]\}$ and $N = |\mathcal{T}| = 2^d - 2$. For $T \in \mathcal{T}$, consider the hidden-spike game $f_T(S) = A\mathbf{1}[S = T]$ with $A > 0$. All these games have the same supplied endpoint values as the zero game. Under the Shapley law, player i's marginal is nonzero only at context $T \setminus \{i\}$, whose probability is

$$p_i(T) = \frac{1}{d\binom{d-1}{|T\setminus\{i\}|}}.$$

Therefore

$$\Delta_i f_T(S) = A\big(2\mathbf{1}[i \in T] - 1\big)\mathbf{1}[S = T \setminus \{i\}],$$
$$\sigma_i^2(f_T) = A^2 p_i(T)(1 - p_i(T)).$$

Since $0 < p_i(T) \leq 1/2$ and $\mathcal{T}$ is finite, choose one amplitude A so that every spike variance exceeds 2ϵ in the additive case. In the multiplicative case, any $A > 0$ suffices. Thus no output valid for the zero game is valid for any spike game: the zero-game valid set is $[-\epsilon, \epsilon]$ or $\{0\}$, respectively.

Run the algorithm on the zero game and let $\mathcal{Q}$ be the set of interior coalitions it queries. Independently choose T uniformly from $\mathcal{T}$ and couple the run on f_T using the same algorithmic randomness. Unless $T \in \mathcal{Q}$, every answer is zero and the adaptive queries, stopping decision, and output coincide. The probability that the outputs differ is consequently at most

$$\Pr(T \in \mathcal{Q}) = \frac{\mathbb{E}[|\mathcal{Q}|]}{N}.$$

The zero-game output belongs to its valid set with probability at least $2/3$. For every spike game, the probability of that same event is at most $1/3$, since its valid outputs lie outside the zero-game valid set. Averaging over T preserves this bound, so the two coupled output laws differ on that event by at least $1/3$. Their probability of disagreement must therefore be at least $1/3$, giving $\mathbb{E}[|\mathcal{Q}|] \geq N/3$. The total query count is at least $|\mathcal{Q}|$, proving the claim. □

G.2 Signed Marginal Moments

Define the signed moments

$$\phi_i^+(f) = \mathbb{E}[\max(\Delta_i f(S), 0)], \qquad \phi_i^-(f) = \mathbb{E}[\min(\Delta_i f(S), 0)],$$

so that $\phi_i(f) = \phi_i^+(f) + \phi_i^-(f)$.

Theorem G.2 (#P-Hardness). *Computing $\phi_i^+(f)$ or $\phi_i^-(f)$ exactly for an ensemble of trees is $\#\mathrm{P}$-hard under polynomial-time Turing reductions.*

Proof. Let nonnegative integers $c_1, \ldots, c_n$ and an integer target k specify a #SubsetSum instance. Set $M = n + 1$ and $c'_j = Mc_j + 1$. For $S \subseteq [n]$, the encoded sum

$$c'(S) = \sum_{j \in S} c'_j = M \sum_{j \in S} c_j + |S|$$

uniquely identifies the original sum and cardinality, since $0 \le |S| < M$. With binary coalition indicators $x \in \{0,1\}^{n+1}$ and a distinguished player i, construct

$$f^\theta(x) = -\theta x_i + \sum_{j=1}^{n} c'_j x_i x_j.$$

The first term is a stump and each remaining term is a depth-two tree. Thus this ensemble has polynomial size and $\Delta_i f^\theta(S) = c'(S) - \theta$. For integer t, the positive moment $H(\theta) = \phi_i^+(f^\theta)$ satisfies

$$H(t-1) - 2H(t) + H(t+1) = \Pr(c'(S) = t).$$

Let $N(k,s)$ count size-s subsets summing to k. Each such coalition has Shapley probability $q_s = s!(n-s)!/(n+1)!$, so

$$\Pr(c'(S) = Mk + s) = q_s N(k,s), \qquad N(k) = \sum_{s=0}^{n} \frac{\Pr(c'(S) = Mk+s)}{q_s}.$$

Hence $3(n+1)$ positive-moment queries recover the #SUBSETSUM count. The integers in the construction and the rational weights have polynomial bit length. Each j belongs to a Shapley context with probability $1/2$, giving

$$\phi_i(f^\theta) = -\theta + \frac{1}{2}\sum_{j=1}^{n} c'_j.$$

Since $\phi_i^+ = \phi_i - \phi_i^-$, a negative-moment oracle also yields the positive moment in polynomial time. □

G.3 CDF AND QUANTILE HARDNESS

Theorem G.3 (Hardness of Exact CDF and Quantile Evaluation)**.** *For marginal contributions under the Shapley coalition distribution, define*

$$F_i(x) = \Pr(\Delta_i f(S) \le x), \qquad Q_i(p) = \inf\{x \in \mathbb{R} : F_i(x) \ge p\}.$$

Exact evaluation of the cumulative distribution function $F_i(x)$ and the quantile function $Q_i(p)$ at rational $p \in (0,1]$ are both #P-hard under polynomial-time Turing reductions, even for ensembles with integer leaf values.

Proof. Use the construction in the proof of Theorem G.2 with $\theta = 0$, so $\Delta_i f(S) = c'(S)$ is integer-valued. Its point masses satisfy

$$\Pr(c'(S) = t) = F_i(t) - F_i(t-1).$$

Substituting these differences into the expression for $N(k)$ gives the #SUBSETSUM count using $2(n+1)$ CDF queries, proving CDF hardness.

To recover the CDF from a quantile oracle, observe that

$$Q_i(p) \le x \quad \Longleftrightarrow \quad p \le F_i(x). \tag{16}$$

If $p \le F_i(x)$, then x belongs to the set defining $Q_i(p)$. Conversely, finite support ensures the infimum is attained, so $Q_i(p) \le x$ implies $p \le F_i(Q_i(p)) \le F_i(x)$. Every coalition probability is an integer multiple of $1/D$ for $D = (n+1)!$, so $F_i(x) = A/D$ for an integer $A \in \{0,\dots,D\}$. Binary search for the largest A satisfying $Q_i(A/D) \le x$, treating $A = 0$ as true without querying the oracle. Equation equation 16 makes this test equivalent to $A/D \le F_i(x)$. The search uses $O(\log D) = O(n \log n)$ queries with polynomial-length rational inputs and returns the exact CDF, proving quantile hardness. □