FASTER APPROXIMATION OF PROBABILISTIC AND DISTRIBUTIONAL VALUES VIA LEAST SQUARES

Weida Li
`vidaslee@gmail.com`

Yaoliang Yu
School of Computer Science
University of Waterloo
Vector Institute
`yaoliang.yu@cs.uwaterloo.ca`

ABSTRACT

The family of probabilistic values, axiomatically-grounded in cooperative game theory, has recently received much attention in data valuation. However, it is often computationally expensive to compute exactly (exponential w.r.t. the number of data being valuated n). The existing generic estimator costs $O(n^2 \log n)$ utility evaluations to achieve an (ϵ, δ)-approximation under the 2-norm, while faster estimators have been developed recently for special cases (e.g., empirically for the Shapley value and theoretically for the Banzhaf value). In this work, starting from the discovered connection between probabilistic values and least square regressions, we propose a **G**eneric **E**stimator based on **L**east **S**quares (GELS) along with its variants that costs $O(n \log n)$ utility evaluations for many probabilistic values, largely extending the scope of this currently best complexity bound. Moreover, we show that each distributional value, proposed by Ghorbani et al. (2020) to alleviate the inconsistency of probabilistic values induced by using distinct databases, can also be cast as optimizing a similar least square regression. This observation leads to a theoretically-grounded framework TrELS (**Tr**aining **E**stimators based on **L**east **S**quares) that can train estimators towards the specified distributional values without requiring any supervised signals. Particularly, the trained estimators is capable of predicting the corresponding distributional values for unseen data, largely saving the budgets required for running Monte-Carlo methods otherwise. Our experiments verify the faster convergence of GELS, and demonstrate the effectiveness of TrELS in learning distributional values. Our code is available at `https://github.com/watml/fastpvalue`.

1 INTRODUCTION

In cooperative game theory, the family of probabilistic values, to which the Shapley value (Shapley 1953) and Banzhaf value (Banzhaf 1965) belong, is uniquely characterized by the axioms of linearity, dummy, monotonicity and symmetry (Weber 1977, Theorems 5 and 10). The induced formula of probabilistic values is often deemed essential for data valuation methods(Kwon and Zou 2022a; Lin et al. 2022; Wang and Jia 2023), and is also proved effective in feature attribution (Jethani et al. 2022; Kwon and Zou 2022b; Lundberg and Lee 2017). Specifically, data valuation aims to impute an importance value to each data point z in the training dataset D_{tr} that represents its contribution to the performance of a model trained on D_{tr}; and it is empirically expected that models retrained without the "less valuable" data (e.g., data that are assigned with importance values lower than a specified threshold) in D_{tr} will achieve performance gains (Ghorbani and Zou 2019).

Throughout, we identify the dataset D_{tr} with $[n] = \{1, 2, \ldots, n\}$, where $n = |D_{tr}|$ is its size. Without ambiguity, for each subset S, its lower-case s is used to denote its cardinality $|S|$, and we write $S \cup i$ and $S \backslash i$ instead of $S \cup \{i\}$ and $S \backslash \{i\}$, respectively. Each probabilistic value can be parameterized by a list of non-negative vectors $\mathcal{P} = \{\mathbf{p}^n \in \mathbb{R}^n_+\}_{n \geq 1}$ such that $\sum_{s=1}^{n} \binom{n-1}{s-1} p^n_s = 1$ for every $n \geq 1$, and the importance value assigned to the i-th data point is computed by

$$\phi_i(U^n) = \phi_i(U^n; \mathcal{P}) = \sum_{S \subseteq [n] \backslash i} p^n_{s+1} \left(U^n(S \cup i) - U^n(S) \right) \tag{1}$$

where $U^n : 2^{[n]} \to \mathbb{R}$ is a user-specified utility function and the superscript n is to indicate its domain $2^{[n]}$. Typically, $U^n(S)$ outputs the performance of a chosen model trained on $S \subseteq [n]$. In this work, $U^n(\emptyset)$ denotes the performance of initialized models. Take classification tasks as example, $U^n(S)$ could be the accuracy or cross entropy loss reported on a held-out dataset D_{perf}. It is obvious that computing $\phi(U^n)$ exactly requires 2^n times of evaluating the utility function U^n, and hence is intractable. Therefore, there has been much research devoted into developing efficient estimators. To our best knowledge, there are only two *generic* estimators designed for all probabilistic values: sampling lift, see Proposition 5, and its weighted variant (Kwon and Zou 2022a). Specifically, the sampling lift estimator requires $O(\frac{n^2}{\epsilon^2} \log \frac{n}{\delta})$ utility evaluations to achieve an (ϵ, δ)-approximation (see Proposition 5 in the Appendix). We also note that Zhou et al. (2023, Lemma 1 and Proposition 1) analyzed convergence using $(\epsilon_1, \epsilon_2, \delta)$-approximation instead where ϵ_1 and ϵ_2 account for the multiplicative and additive error, respectively.

So far, many faster estimators have been proposed for specific probabilistic values. For instance, the advent of faster estimators designed specifically for the Shapley value was witnessed (Covert and Lee 2021; Kolpaczki et al. 2023; Lundberg and Lee 2017; Zhang et al. 2023b); Wang and Jia (2023, Theorem 4.9) proved that the estimator based on their proposed maximum sample reuse (MSR) principle only requires $O(\frac{n}{\epsilon^2} \log \frac{n}{\delta})$ utility evaluations for the Banzhaf value, but they also demonstrated in Appendix C.2 therein that the MSR estimator does not extend to many other probabilistic values, e.g., the family of Beta Shapley values (Kwon and Zou 2022a), in which the Shapley value resides. We also notice that Lin et al. (2022) discovered a framework of unconstrained least square regressions that leads to an estimator for a subfamily of probabilistic values (which, however, does not include the Shapley value). All in all, it is still an open question on how to efficiently approximate other probabilistic values, e.g., the Beta Shapley values.

A potential drawback of the probabilistic values is that they depend on the underlying database D_{tr}. Using another database, the recalculated importance values could be very inconsistent with the previous ones. To overcome this issue, Ghorbani et al. (2020) proposed the framework of distributional values, in which the importance value of any (unseen) data point z is

$$\psi(z; \mathcal{D}, \mathbf{w}, U) = \mathop{\mathbb{E}}_{s \overset{\mathbf{w}}{\sim} [m]} \mathop{\mathbb{E}}_{S \sim \mathcal{D}^{s-1}} [U(S \cup z) - U(S)] \tag{2}$$

where $\mathcal{D}$ is a data distribution, $\mathbf{w} \in \mathbb{R}^m$ is a probability vector and the domain of the utility function U is $\bigcup_{n \geq 1} \{S \sim \mathcal{D}^n\}$. Empirically, $\mathcal{D}$ is replaced by $\frac{1}{|B|} \sum_{z \in B} \delta_z$ where B is a (large) dataset sampled from $\mathcal{D}$ (Ghorbani et al. 2020). Clearly, the computational cost for the distributional values is a big hurdle for its practical deployment.

In this paper, starting from the discovered connection between probabilistic values and least squares (see Proposition 2), we develop a **G**eneric **E**stimator based on **L**east **S**quares (GELS) along with its variants: GELS-R and GELS-Shapley. Precisely, GELS-R is to approximate the ranking of the specified probabilistic values instead, while GELS-Shapley is specific for the Shapley value. In addition, we demonstrate that each distributional value can also be cast into a similar least square regression, which serves as the theoretical ground for formulating the framework TrELS (**Tr**aining **E**stimators based **L**east **S**quares) that can train estimator towards the specified distributional values without requiring any supervised signals. In other words, unlike Monte-Carlo methods, TrELS allows the use of trained estimator for prediction. Our main contributions are summarized as follows:

1. We propose GELS and GELS-R for all probabilistic values, and prove that both of them requires $O(\frac{n}{\epsilon^2} \log \frac{n}{\delta})$ utility evaluations to achieve an (ϵ, δ)-approximation for many probabilistic values, matching the currently best bound for special cases. See Algorithms 1 and 2, Propositions 1, 3 and 4. In addition, we also develop GELS-Shapley specific for the Shapely value.
2. By casting the distributional values into optimizing least square regressions, we design TrELS for training estimators towards distributional values without supervision. See Algorithm 4 and Theorem 1.
3. Our experiments verify the faster convergence of GELS and GELS-R, and we also show that the distributional values can be well-learned using TrELS.
4. As a minor side note, we also extend the approximation-without-requiring-marginal (ARM) estimator, designed specifically for the Shapley value (Kolpaczki et al. 2023), to the whole family of probabilistic values. See Proposition 8 in the Appendix.

2 BACKGROUND

The sampling lift estimator (Moehle et al. 2022) refers to any approximation algorithm designed according to

$$\phi_i(U^n) = \sum_{S\subseteq[n]\backslash i} p_{s+1}^n \left(U^n(S\cup i) - U^n(S)\right) = \mathbb{E}_{S\subseteq[n]\backslash i}[U^n(S\cup i) - U^n(S)].$$

Its weighted variant is to use $n\binom{n-1}{s-1}p_{s+1}^n(U^n(S\cup i) - U^n(S))$ with $P(S) = \frac{1}{n}\binom{n-1}{s}^{-1}$ instead Kwon and Zou (2022a). Particularly, they are the same for the Shapley value. As summarized in Proposition 5, the sampling lift requires $O(\frac{n^2}{\epsilon^2}\log\frac{n}{\delta})$ utility evaluations to achieve an (ϵ,δ)-approximation. Recently, Kolpaczki et al. (2023) proposed an ARM estimator specifically for the Shapley value, but we notice that it can be easily generalized for all probabilistic values. Therefore, we present the *generic* formula of ARM in the main paper while leaving its justification to Proposition 8 in the Appendix. The ARM estimator is to approximate by using

$$\phi_i(U^n) = \mathbb{E}_{S\sim P_{ARM}^{+}\mid i\in S}[U^n(S)] - \mathbb{E}_{S\sim P_{ARM}^{-}\mid i\not\in S}[U^n(S)]$$

where $P_{ARM}^{+}(S)\propto p_s^n$ for all *non-empty* subsets $S\subseteq[n]$ and $P_{ARM}^{-}(S)\propto p_{s+1}^n$ for all $S\subsetneq[n]$.

In addition, Lin et al. (2022) developed the average marginal effect (AME) that casts a *subfamily* of probabilistic values as the uniquely optimal solution to

$$\operatorname*{arg\,min}_{\mathbf{v}\in\mathbb{R}^n} \mathbb{E}[(Y - \boldsymbol{X}^\top\mathbf{v})^2],$$

where $\boldsymbol{X}$ and Y are random variables whose distribution is induced by i) sampling $t\sim P$ where P is any probability distribution on the open interval $(0,1)$, ii) sampling a subset $S\subseteq[n]$ by including each data point in $[n]$ with probability t, and iii) setting $Y = U^n(S)$ and $X_i = \frac{1}{t\cdot M_P}$ if $i\in S$ and $\frac{-1}{(1-t)M_P}$ otherwise where $M_P = \mathbb{E}_{t\sim P}[\frac{1}{t(1-t)}]$. Particularly, the uniquely optimal solution just a probabilistic value parameterized by $p_s^n = \int_0^1 t^{s-1}(1-t)^{n-s}dP(t)$. If P is uniform, the corresponding $\mathbf{p}^n$ is the one defining the Shapley value, but $M_P = \infty$, and thus AME does not work for the Shapley value.

On the other hand, Wang and Jia (2023) proposed the maximum sample reuse (MSR) principle which aims to find a distribution P_{MSR} on $2^{[n]}$ such that for every $i\in[n]$,

$$P_{MSR}(S\mid i\in S) = p_s^n \text{ and } P_{MSR}(S\mid i\not\in S) = p_{s+1}^n,$$

with which Eq. (1) can be rewritten as $\phi_i(U^n) = \mathbb{E}_{S\mid i\in S}[U^n(S)] - \mathbb{E}_{S\mid i\not\in S}[U^n(S)]$. Specifically, P_{MSR} is uniform over $2^{[n]}$ for the Banzhaf value, and Wang and Jia (2023, Theorem 4.9) showed that the resulting estimator only requires $O(\frac{n}{\epsilon^2}\log\frac{n}{\delta})$ utility evaluations. However, they also demonstrated in Appendix C.2 therein that such P_{MSR} exists if and only if $p_{s+1}^n = \eta\cdot p_s^n$ for every $1\leq s\leq n-1$ (n is treated as fixed) where $\eta\in\mathbb{R}$, which excludes all Beta Shapley values.

A recently-proposed faster algorithm specific for the Shapley value is the complement estimator (Zhang et al. 2023b) using the complement formula

$$\phi_i^{Sh}(U^n) = \sum_{S\subseteq[n]:\, i\in S} \frac{s!(n-1-s)!}{n!}\left(U^n(S) - U^n([n]\backslash S)\right),$$

which was discovered half a century ago by Harsanyi (1963, Eq. (4.1)). Earlier, Lundberg and Lee (2017) proposed the kernelSHAP estimator that exploits the fact that the Shapley value $\phi^{Sh}(U^n)$ is the unique optimal solution to

$$\begin{aligned}\operatorname*{arg\,min}_{\mathbf{v}\in\mathbb{R}^n} \sum_{\emptyset\subsetneq S\subsetneq[n]} \binom{n-2}{s-1}^{-1}\left(U^n(S) - U^n(\emptyset) - \sum_{i\in S} v_i\right)^2 \\ \text{s.t.} \sum_{i\in[n]} v_i = U^n([n]) - U^n(\emptyset),\end{aligned} \tag{3}$$

which was first discovered by Charnes et al. (1988, Theorem 4). Since it is difficult to analyze whether the kernelSHAP is unbiased or not, Covert and Lee (2021, Eq. (9)) later proposed a (provably) unbiased variant, and suggested that the paired sampling technique can enhance both. Very recently, Fumagalli et al. (2023, Theorem 4.5) and Zhang et al. (2023a, Eqs. (11) and (12)) simplified the formula of the unbiased kernelSHAP.

3 MAIN RESULTS

In this section, we first show how to efficiently estimate the ranking induced by any probabilistic value. Then, by introducing a null data point we show how to turn our ranking estimator into a *bona fide* estimator, while retaining the same efficiency for many probabilistic values. Lastly, we extend our theory to an unsupervised framework that trains estimators towards distributional values.

3.1 WHEN RANKING SUFFICES

Practitioners often need to screen training data before feeding them to a model, to remove outliers, low-quality data or even adversarial examples that are deemed harmful to training. Data valuation is a natural way to achieve this goal, i.e., only data assigned with high importance values could be considered potentially "valuable." If one has a good estimate of the proportion of "valuable" data, then the relative ranking, instead of the more precise and demanding importance values, suffices. Our first results below pave the way to efficiently estimate the relative ranking underlying any probabilistic value. Particularly, $\mathcal{G}^n = \{U^n : 2^{[n]} \to \mathbb{R}\}$ is the set that contains all possible utility functions provided there are n data to valuate.

Proposition 1. *Define, for every $n \geq 1$, $U^n \in \mathcal{G}^n$ and $i \in [n]$,*

$$\mathcal{R}_i(U^n) = \mathcal{R}_i(U^n; \mathcal{P}) = \sum_{\emptyset \subsetneq S \subsetneq [n]} m_s^n \cdot \mathbb{1}_{i \in S} U^n(S) \tag{4}$$

where $m_s^n = m_s^n(\mathcal{P}) = p_s^n + p_{s+1}^n$ for every $s \in [n-1]$. Then, for every $n \geq 1$ and $U^n \in \mathcal{G}^n$, $\mathcal{R}(U^n)$ and $\phi(U^n)$ produce the same ranking. Precisely, $\mathcal{R}(U^n) = \phi(U^n) + g(U^n)\mathbf{1}_n$ where $g(U^n) = g(U^n; \mathcal{P}) \in \mathbb{R}$. As a side note, it holds true for any $\mathbf{p}^n \in \mathbb{R}^n$.

We emphasize that Proposition 1 is closely related to Proposition 2 below in the sense that they immediately imply each other; we *first* noticed a more general version of proposition 2 that applies to all least square values (Ruiz et al. 1998, Definition 5), a broader family that includes all the additive-efficient-normalized probabilistic values; see Appendix D for more details.

Proposition 2. *Consider the problem*

$$\mathop{\arg\min}_{\mathbf{v} \in \mathbb{R}^n} \sum_{\emptyset \subsetneq S \subsetneq [n]} m_s^n \cdot \Big(U^n(S) - \sum_{i \in S} v_i\Big)^2, \tag{5}$$

its uniquely optimal solution shares the same ranking as $\phi(U^n)$ for every $n \geq 1$ and $U^n \in \mathcal{G}^n$. As a side note, it holds true for any $\mathbf{p}^n \in \mathbb{R}^n$ that produces non-negative weight vector $\mathbf{m}^n$ with $\sum_{s=1}^{n-1} m_s^n > 0$.

Algorithm 1 summarizes the estimator induced by Propositions 1 and 2. Note that for GELS-R each utility evaluation $U^n(S)$ can be used for updating the estimate probabilistic values of s data. By contrast, the (weighted) sampling lift estimator spends two utility evaluations to update the estimate of only one data point.

3.2 WHEN PROBABILISTIC VALUES ARE DESIRED

Remark 1. *To recover $\phi(U^n)$ from $\mathcal{R}(U^n)$, we introduce a null data point labeled as $n+1$ to extend each U^n into $\overline{U}^{n+1}$ such that $\overline{U}^{n+1}(S \cup n+1) = U^n(S)$ for every $S \subseteq [n]$. Meanwhile, we construct $\mathbf{p}^{n+1} \in \mathbb{R}^{n+1}$ such that $p_s^n = p_s^{n+1} + p_{s+1}^{n+1}$ for every $s \in [n]$, and note that $\mathbf{p}^{n+1}$ may contain negative weights. Proposition 1 demonstrates that $\mathcal{R}(\overline{U}^{n+1}) = \phi(\overline{U}^{n+1}) + g(\overline{U}^{n+1})\mathbf{1}_{n+1}$. Particularly, the definition of Eq. (1) makes that $\phi_{n+1}(\overline{U}^{n+1}) = 0$ and the structure $p_s^n = p_s^{n+1} + p_{s+1}^{n+1}$ leads to $\phi_i(U^n) = \phi_i(\overline{U}^{n+1})$ for every $i \in [n]$. Therefore, $g(\overline{U}^{n+1}) = \mathcal{R}_{n+1}(\overline{U}^{n+1})$, and we have the recovery formula $\phi_i(U^n) = \phi_i(\overline{U}^{n+1}) = \mathcal{R}_i(\overline{U}^{n+1}) - \mathcal{R}_{n+1}(\overline{U}^{n+1})$. This idea is summarized in Proposition 3 and Algorithm 2.*

Proposition 3. *The unique optimal solution $\mathbf{v}^*$ to the problem*

$$\mathop{\arg\min}_{\mathbf{v} \in \mathbb{R}^{n+1}} \sum_{\emptyset \subsetneq S \subsetneq [n+1]} p_s^n \cdot \Big(U^n(S \cap [n]) - \sum_{i \in S} v_i\Big)^2 \tag{6}$$

Algorithm 1: GELS-R (**G**eneric **E**stimator based on **L**east **S**quares for **R**ankings)

```
Input: A dataset $D_{tr} \equiv [n]$ to be valuated, a utility function $U^n \in \mathcal{G}^n$, a weight vector
       $\mathbf{q} \in \mathbb{R}^{n-1}$ defined by $q_s = \binom{n}{s}(p_s^n + p_{s+1}^n)$, and the total number $T$ of samples
Output: An unbiased estimate $\hat{\mathbf{r}}$ to $\mathcal{R}(U^n)$ up to some scalar
Normalize $\mathbf{w}$ into a probability vector $\mathbf{q} \leftarrow \mathbf{q}/\sum_{s=1}^{n-1} q_s$
$\hat{\mathbf{r}} \leftarrow \mathbf{0}_n, \mathbf{t} \leftarrow \mathbf{0}_n$
for $k = 1, 2, \ldots, T$ do
    Sample $s_k \in [n-1]$ using the probability vector $\mathbf{q}$
    Uniformly sample $S_k$ from $\{R \subseteq [n] \mid |R| = s_k\}$
    for $i \in S_k$ do
        $t_i \leftarrow t_i + 1$
        $\hat{r}_i \leftarrow (1 - \frac{1}{t_i})\hat{r}_i + \frac{1}{t_i} U^n(S_k)$
```

Algorithm 2: GELS (**G**eneric) **E**stimator based on **L**east **S**quares

```
Input: A $D_{tr} \equiv [N]$ to be valuated, a utility function $U^n \in \mathcal{G}^n$, a weight vector $\mathbf{q} \in \mathbb{R}^n$
       defined by $q_s = \binom{n+1}{s} p_s^n$, and the total number $T$ of samples
Output: An unbiased estimate $\hat{\boldsymbol{\phi}}$ to $\phi(U^n)$
Introduce a null datum labeled by $n+1$, and extend $U^n$ to $\overline{U}^{n+1}$   // $\overline{U}^{n+1}(S) = U^n(S \cap [n])$
Obtain $\hat{\mathbf{r}} \in \mathbb{R}^{n+1}$ from Algorithm 1 using $[n+1]$, $\overline{U}^{n+1}$, $\mathbf{q}$ and $T$
$\hat{\phi}_i \leftarrow (\sum_{s=1}^{n} \frac{s}{n+1} q_s)(\hat{r}_i - \hat{r}_{n+1})$ for each $i \in [n]$   // using unnormalized $\mathbf{q}$
```

satisfies that $\phi_i(U^n) = v_i^* - v_{n+1}^*$ *for every* $i \in [n]$.

Remark 2. *For the Shapley value, since* $\sum_{i\in[n]} \phi_i^{Sh}(U^n) = U^n([n]) - U^n(\emptyset)$ *and by Proposition 1* $\phi(U^n) = \mathcal{R}(U^n) + g(U^n)\mathbf{1}_n$*, we have* $U^n([n]) - U^n(\emptyset) = \sum_{i\in[n]} \mathcal{R}_i(U^n) + g(U^n) \cdot n$*, and thus* $g(U^n) = \frac{1}{n}(U^n([n]) - U^n(\emptyset) - \sum_{i\in[n]} \mathcal{R}_i(U^n))$*. Therefore, we can recover* $\phi^{Sh}(U^n)$ *without introducing a null data point, which is summarized in Algorithm 3.*

Next, we present convergence analysis for GELS, GELS-R and GELS-Shapley.

Proposition 4. *Assume that* $\|U^n\|_\infty \leq u$ *for every* $U^n \in \bigcup_{n\geq 1} \mathcal{G}^n$*. GELS and GELS-Shapley estimators require* $T = O(\frac{\tau(n)n}{\epsilon^2} \log \frac{n}{\delta})$ *samples, which is equal to the number of utility evaluations, to achieve* $P(\|\hat{\phi}(U^n) - \phi(U^n)\| \geq \epsilon) \leq \delta$*. For GELS-R estimator, it requires* $T = O(\frac{\kappa(n)n}{\epsilon^2} \log \frac{n}{\delta})$ *samples (equivalently, utility evaluations) instead to achieve an* (ϵ, δ)*-approximation.*

Remark 3. *In Appendix H, we demonstrate that both* $\tau(n)$ *and* $\kappa(n)$ *is the inverse square of the average reuse rate of utility evaluations. In other words, the more estimates in* $\hat{\boldsymbol{\phi}}$ *each utility function is used to update, the more efficient the GELS and its variants will be. Precisely, for semi-values, which include all probabilistic values mentioned and whose* $\mathcal{P}$ *can be summarized by a probability measure* μ *over the closed interval* $[0,1]$ *by* $p_s^n = \int_0^1 t^{s-1}(1-t)^{n-s} \mathrm{d}\mu(t)$*, we prove that* $\tau(n), \kappa(n) \in \Theta(1)$ *if* $\int_0^1 \frac{1}{t(1-t)} \mathrm{d}\mu(t) < \infty$*. To sum, the Banzhaf value and the Beta Shapley values with* $\alpha, \beta > 1$ *satisfies this condition; for the Beta Shapley values with* $\alpha, \beta \geq 1$*, it becomes* $\tau(n), \kappa(n) \in O(\log^2 n)$ *instead; particularly,* $\kappa(n) \in \Theta(1)$ *for the Shapley value, which indicates GELS-R requires* $O(\frac{n}{\epsilon^2} \log \frac{n}{\delta})$ *utility evaluations to achieve an* (ϵ, δ)*-approximation. We refer the reader to Appendices F and I for more details. By contrast, the sampling lift estimator always requires* $O(\frac{n^2}{\epsilon^2} \log \frac{n}{\delta})$ *utility evaluations to achieve an* (ϵ, δ)*-approximation.*

3.3 DISTRIBUTIONAL VALUE

We are now ready to combine the previous results with the notion of distributional values Eq. (2) to establish an unsupervised framework for *training* estimators towards distributional values; see Appendix A for more details. Such trained estimators can valuate any unseen data point sampled from the same (or close-enough) data distribution in a single forward pass, which largely saves the

Algorithm 3: GELS-Shapley (**G**eneric) **E**stimator based on **L**east **S**quares for the **S**hapley value

```
Input: A $D_{tr} \equiv [N]$ to be valuated, a utility function $U^n \in \mathcal{G}^n$, a weight vector $\mathbf{q} \in \mathbb{R}^{n-1}$
       defined by $q_s = \frac{n}{s(n-s)}$, and the total number $T$ of samples
Output: An unbiased estimate $\hat{\boldsymbol{\phi}}$ to the Shpaley value of $U^n$
Obtain $\hat{\mathbf{r}} \in \mathbb{R}^n$ from Algorithm 1 using $[n]$, $\overline{U}^n$, $\mathbf{q}$ and $T$
$\hat{r}_i \leftarrow \hat{r}_i \cdot H_{n-1}$ for each $i \in [n]$                // $H_{n-1} = \sum_{s=1}^{n-1} \frac{1}{s}$
$\hat{\phi}_i \leftarrow \hat{r}_i + O$ for each $i \in [n]$        // $O = (U^n([n]) - U^n(\emptyset) - \sum_{i=1}^{n} \hat{r}_i)/n$
```

Algorithm 4: TrELS (**Tr**aining **E**stimators based on **L**east **S**quares)

```
Input: A database $D_{tr} \equiv (\mathbf{x}_i, y_i)_{1 \le i \le n}$, a probability vector $\boldsymbol{\omega} \in \mathbb{R}^m$ with $m < n$, a utility
       function $U^n \in \mathcal{G}^n$, a trainable model $\phi_{\boldsymbol{\theta}}$ with an additional trainable parameter $\phi_o$, the
       batch size $Z$ and the total number $T$ of batches used for training.
Compute $\mathbf{d} \in \mathbb{R}^m$ by letting $d_s = \frac{n-s+1}{s}\omega_s$ for every $s \in [m]$
Normalize: $\mathbf{d} \leftarrow \mathbf{d}/\sum_{s=1}^{m} d_s$
for $t = 1, 2, \ldots, T$ do
    $loss \leftarrow 0$
    for $j = 1, 2, \ldots, Z$ do
        Sample $s_j$ from $[m]$ according to the distribution vector $\mathbf{d}$
        Sample $S_j$ uniformly from $\{R \subseteq [n+1] \mid |R| = s_j\}$
        $loss \leftarrow loss + \left(U^n(S_j \cap [n]) - \phi_o \mathbb{1}_{n+1 \in S_j} - \sum_{i \in S_j \cap [n]} \phi_\theta(\mathbf{x}_i, y_i)\right)^2$
    $loss \leftarrow loss/Z$
    Update $\boldsymbol{\theta}$ and $\phi_o$ using $loss$
Evaluation Phase: $C \cdot (\phi_{\boldsymbol{\theta}}(\mathbf{x}, y) - \phi_o)$                // $C = \sum_{s=1}^{m} \frac{n-s+1}{n}\omega_s$
```

budgets required for running Monte-Carlo methods otherwise. As counterparts in feature attribution, Covert et al. (2023) and Jethani et al. (2022) exploited the least square regression (3) together with the additive efficient normalization proposed by Ruiz et al. (1998, Definition 11) to train neural networks that can then predict the Shapley value of any unseen instance in a single forward pass.

In practice, the data distribution $\mathcal{D}$ in Eq. (2) is replaced by an empirical distribution $\mathcal{B} = \frac{1}{|B|}\sum_{z \in B} \delta_z$ where B is a sufficiently large dataset sampled by $\mathcal{D}$. Substituting $\mathcal{B}$ for $\mathcal{D}$ in Eq. (2), observe that there exists another probability vector $\boldsymbol{\omega} \in \mathbb{R}^m$ such that

$$\psi(z; \mathcal{B}, \mathbf{w}, U) = \varphi(z; \mathcal{B}, \boldsymbol{\omega}, U) := \mathop{\mathbb{E}}_{s \overset{\boldsymbol{\omega}}{\sim} [m]} \; \mathop{\mathbb{E}}_{S \overset{\mathcal{U}}{\sim} \mathcal{B}_{s-1}} [U(S \cup z) - U(S)] \tag{7}$$

where $\mathcal{U}$ represent uniform sampling and $\mathcal{B}_{s-1} = \{R \subseteq B \mid |R| = s-1\}$. We point out φ is more natural to analyze. If $z \notin B$, Eq. (7) is equal to calculating some probabilistic value for z with domain of utility function being $2^{B \cup z}$; if $z \in B$, it produces some probabilistic value of z up to some scalar, with the domain of utility function being 2^B.

Theorem 1. *Consider the case when $B = [n]$ (recall that $D_{tr} \equiv [n]$) for the empirical distributional values φ defined in Eq. (7). Let $\mathbf{d} \in \mathbb{R}^m$ be a probability vector that satisfies $d_s \propto \frac{n-s+1}{s}\omega_s$ for every $s \in [m]$, and $\mathbf{v}^*$ be the uniquely optimal solution to*

$$\mathop{\arg\min}_{\mathbf{v} \in \mathbb{R}^{n+1}} \; \mathbb{E}_{s \overset{\mathbf{d}}{\sim} [m]} \, \mathbb{E}_{S \overset{\mathcal{U}}{\sim} \overline{\mathcal{B}}_s} \left(U^n(S \cap [n]) - \sum_{i \in S} v_i\right)^2, \tag{8}$$

where $\overline{\mathcal{B}}_s = \{R \subseteq [n+1] \mid |R| = s\}$. There is, for every $i \in [n]$,

$$C \cdot (v_i^* - v_{n+1}^*) = \varphi(i; \mathcal{B}, \boldsymbol{\omega}, U^n)$$

where $C = \sum_{s=1}^{m} \frac{n-s+1}{n}\omega_s$. As a side note, $\hat{\mathbf{r}} \in \mathbb{R}^{n+1}$ obtained in Algorithm 2 (substituting $\boldsymbol{\omega}$ for $\mathbf{p}^n$ while letting $\omega_s = 0$ if $s > m$) satisfies that $\mathbb{E}[\hat{r}_i] - \mathbb{E}[\hat{r}_{n+1}] = \varphi(i; \mathcal{B}, \boldsymbol{\omega}, U^n)$ for every $i \in [n]$.

Theorem 1 constitutes a theoretically unsupervised ground for training estimators towards distributional values. Precisely, let $\phi_{\boldsymbol{\theta}}(\mathbf{x}, y)$ be a trainable model parameterized by $\boldsymbol{\theta}$; $\mathbf{x}$ and y stands for features and label, respectively; we propose training the estimator $\phi_{\boldsymbol{\theta}}$ based on the specified utility function $U^n \in \mathcal{G}^n$ through optimizing

$$\mathop{\arg\min}_{\boldsymbol{\theta},\phi_o} \mathop{\mathbb{E}}_{s \overset{\mathbf{d}}{\sim} [m]} \mathop{\mathbb{E}}_{S \overset{\mathcal{U}}{\sim} \overline{\mathcal{B}}_s} \left(U(S \cap [n]) - \phi_o \mathbb{1}_{n+1 \in S} - \sum_{i \in S \setminus n+1} \phi_{\boldsymbol{\theta}}(\mathbf{x}_i, y_i) \right)^2 . \tag{9}$$

As will be seen in the experiments, for any data point $z \notin B$, the trained $\phi_{\boldsymbol{\theta}}$ indeed predicts quite accurately the corresponding probabilistic value of z with the domain of utility function being $2^{B \cup z}$!

Generally, it is impractical to train estimators using the exact distributional values, or regressing with a reasonable approximation as supervised signals, since it is extremely expensive to calculate them exactly. For example, in our experiments where $N = 10,000$ and $m = 1,000$, each unseen data point requires $\sum_{s=0}^{999} \binom{10,000}{s}$ utility evaluations to compute exactly. Therefore, the merits of training estimators through optimizing the problem (9) are clear: i) we do not have to be concerned with what the exact distributional values are, and ii) Theorem 1 guarantees that each estimator will be trained towards the exact distributional values. The outline of such an unsupervised framework is summarized in Algorithm 4.

4 EVALUATION

In this section we perform experiments to i) verify the faster convergence of Algorithms 1 and 2, and ii) demonstrate that distributional values can be effectively predicted by trained estimators using TrELS. The classification datasets employed are from open resources, which are iris, wind (both are from OpenML), FMNIST (Xiao et al. 2017) and MNIST. For all utility functions, their outputs are set to be the performance of the trained models reported on a dataset D_{perf} disjoint from D_{tr}. Take U^n as example, $U^n(S)$ reports the performance of the specified model trained on $S \subseteq D_{tr} \equiv [n]$, and we leave the specified model untrained for evaluating $U^n(\emptyset)$. Without stated explicitly, the performance of trained models is measured by classification accuracy. In addition, we fix the random seed to be 2024 to make all utility functions *deterministic*. To be computationally efficient, we adopt one-mini-batch one-epoch learning for evaluating utility functions (Ghorbani and Zou 2019). Logistic regression is implemented for iris and wind, while LeNet (LeCun et al. 1998) is employed for FMNIST and MNIST.

4.1 FASTER CONVERGENCE OF THE PROPOSED ESTIMATORS

This experiment is performed on all above-mentioned datasets. The SGD optimizer with learning rate 1.0 is employed for iris and wind, whereas we set the learning rate to be 0.1 instead for MNIST and FMNIST. To evaluate the specified probabilistic values exactly, we set $|D_{tr}| = |D_{perf}| = 24$. All estimators are run 30 times with different random seeds, which range from 0 to 29. We compare to the following benchmarks:

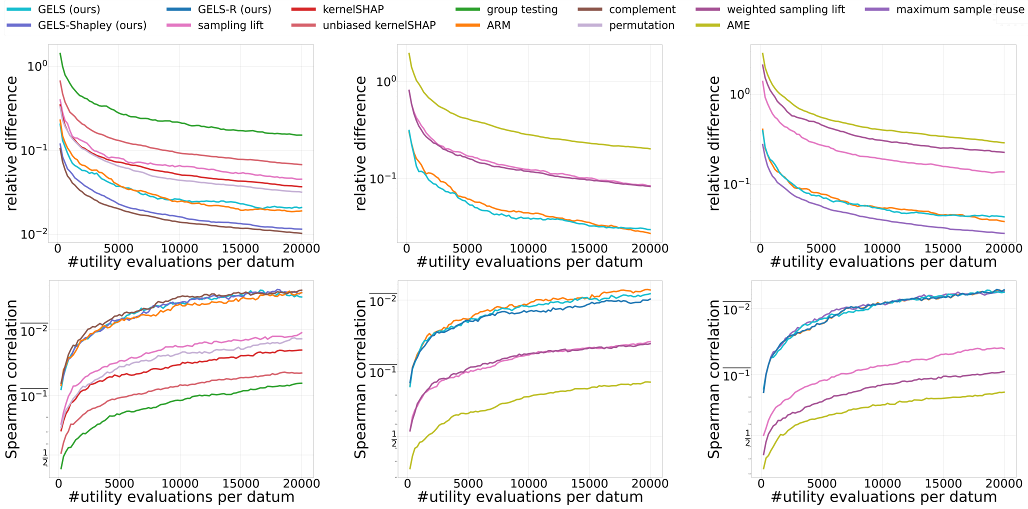
- those designed specifically for the Shapley value, including the complement (Zhang et al. 2023b), kernelSHAP (Lundberg and Lee 2017), unbiased KernelSHAP (Covert and Lee 2021), group testing (Jia et al. 2019);
- those work for all probabilistic values, including the sampling lift (see Proposition 5) and its weighted variant (Kwon and Zou 2022a), and the generalized ARM (see Proposition 8);
- AME (Lin et al. 2022), restricted to a subfamily of probabilistic values;
- MSR (Wang and Jia 2023), proposed specifically for the Banzhaf value.

The probabilistic values we consider are: i) Beta$(1, 1)$, which equals to the Shapley value, ii) Beta$(2, 2)$,[1] iii) Beta(4, 1), and iv) the Banzhaf value. Particularly, AME does not apply to Beta(4, 1) and the Shapley since both of them lead to $M_P = \infty$.

Remark 4. *Note that Covert and Lee (2021) proposed the paired sampling technique to enhance (unbiased) kernelSHAP. We point out that this technique can also be employed for our methods, (weighted) sampling lift, group testing, AME, and MSR. To be fair, we only compare the plain estimators in the main paper. Experiments including the paired sampling are included in Appendix J.*

---

[1] This probabilistic value is special: i) it can be derived from the best min-polynomial approximation of the Lovász extension of U^n (Marichal and Mathonet 2008, Corollary 19), and ii) it is uniquely characterized by a set of axioms proposed in multi-criteria decision making (Grabisch and Labreuche 2001, Theorem 2)



Shapley Beta(2, 2) Banzhaf

Figure 1: Comparison of different estimators on probabilistic values using wind. The relative difference is plotted in log scale while the Spearman correlation is in logit scale. In addition, $\overline{r} = 1 - r$.

We present some of the results in Figure 1, while defer others to Appendix J. These results support that GELS, GELS-R are currently among the (empirically) fastest tier in the group of *generic* estimators. For the Shapley value, we omit the results of GELS-R since GELS-R and GELS-Shapley share the ranking; GELS-Shapley converges significantly faster than GELS; however, GELS-Shapley may converge slower compared with the complement and kernelSHPA; notice that their faster convergence comes at the cost of using $\Theta(n^2)$ memory storage instead of $\Theta(n)$; precisely, the complement requires another $O(n^2)$ time complexity to aggregate all n^2 estimates, whereas kernelSHAP needs an extra $O(n^3)$ time complexity for calculation the inverse of matrices. For the Banzhaf value, though GELS converges slower than MSR in terms of relative error, the performance of GELS and GELS-ranking is not worse than MSR, and sometimes better, in terms of Spearman correlation. For the Beta$(2, 2)$, GELS and GELS-R achieve the best. We notice that the generalize ARM almost enjoys the same empirical convergence across different experiment settings, except that it is slower than GELS-Shapley in terms of relative error.

4.2 TRAINING ESTIMATORS

In this experiment, we demonstrate the effectiveness of training estimators using Algorithm 4 on FMNIST and MNIST. The corresponding results for MNIST are included in Appendix J. For the utility functions, the SGD optimizer with learning rate 0.01 and one-mini-batch one-epoch learning is adopted, and we take $|D_{tr}| = 10,000$ and $|D_{perf}| = 500$ from the training set. LeNet is the architecture we employ as a trainable estimator $\phi_{\boldsymbol{\theta}}$, and the input of the softmax layer is taken as the output of $\phi_{\boldsymbol{\theta}}$. For training estimators, we employ the Adam optimizer (Kingma and Ba 2014) with learning rate 0.001. The batch size Z is set to be $10,000$, and we randomly generate $1,000$ batches. After these batches have all been fed into the estimators, we permute and reuse them to continue training. Therefore, we have in total $1,000$ utility evaluations per data point for training estimators. Lastly, the probability vector $\boldsymbol{\omega} \in \mathbb{R}^{1,000}$ we employ is $\omega_s \propto s^{-\frac{1}{2}}$. Moreover, 200 data, denoted by D_{val}, are taken from the training dataset for selecting the best trained models. For the results in the second row of Figure 2, we report on another 200 data, denoted by D_{test}, extracted from the test dataset. To sum, D_{tr}, D_{perf}, D_{val} and D_{test} are all disjoint, which means D_{val} and D_{test} are composed of unseen data. For D_{val} and D_{test}, we randomly run $600,000$ utility evaluations for each data point using Eq. (7) to generate estimates of the specified distributional values, which are regarded as the ground-truths $\boldsymbol{\varphi}$. All results are reported with mean and standard deviation using 30 different random seeds ranging from 0 to 29.

The performance curves of the estimators during training are shown in Figure 2. First of all, the increasing curve in terms of the Spearman correlation indicates that the estimators trained under TrELS are able to gradually learn the exact ranking. Secondly, the decreasing curve (the dashsed

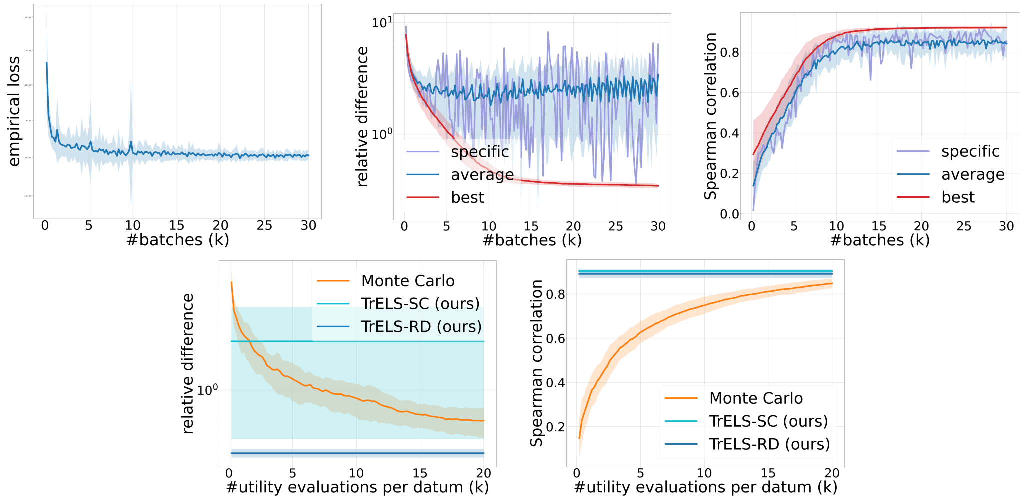


Figure 2: i) The first row: The first plot is the empirical loss of the sampled batch, while the others present the relative distance $\|\hat{\varphi}-\varphi\|_2/\|\varphi\|_2$ and the Spearman correlation between φ and $\hat{\varphi}$. $\hat{\varphi}$ denotes the one predicted by the estimators trained on FMNIST. The first two plots are in log scale. "best" reports the best one achieved by the trained estimators at each time point, whereas "specific" is the one with the random seed being 0. ii) The second row: the two plots compare the performance of the estimators trained on FMNIST with Monte-Carlo method using Eq (2). Specifically, TrELS-SC uses trained estimators that achieve the best Spearman correlation on D_{val}, whereas RD stands for relative difference.

line) of the relative difference suggests that the transform $C \cdot (\phi_{\boldsymbol{\theta}}(\mathbf{x}, y) - \phi_o)$ in Algorithm 4 is essential for training!

Next, we examine the efficiency of the trained estimators by comparing them with Monte-Carlo method based on Eq. (7). Specifically, for each random seed, the trained estimators that achieves the best performance on D_{val} are selected. The comparison is shown in the second row of Figure 2. As clearly shown, the Monte-Carlo methods using $20,000$ utility evaluations are still inferior to the estimators trained under our TrELS. Observe that TrELS-SC performs poorly in terms of relative difference, which is expected since the reported relative differences of estimators during training are significantly unstable.

5 CONCLUSION

In this work, we start from the least square regression to develop GELS with its variants for all probabilistic values. GELS-R is to approximate the ranking of probabilistic values, whereas GELS-Shapley is specific designed for the Shapley value. The faster convergence of the proposed estimators is theoretically guaranteed, and is also verified empirically in our experiments. Besides, we also demonstrate how to cast each distributional value into a least square problem, making it the first-time theoretically-grounded to train estimators towards distributional values in an unsupervised manner, the framework of which is introduced as TrELS. Notably, our experiments show that the estimators trained under TrELS learn the specified distributional values quite well in terms of both relative difference and Spearman correlation. Our work significantly broadens the practicality to deploy value-based data valuation methods on rather large datasets.

ACKNOWLEDGEMENTS

We thank the reviewers and the area chair for thoughtful comments that have improved our final presentation. YY gratefully acknowledges NSERC and CIFAR for funding support.

REFERENCES

Banzhaf III, J. F. (1965). "Weighted Voting Doesn't Work: A Mathematical Analysis". *Rutgers Law Review*, vol. 19, pp. 317–343.

Charnes, A., B. Golany, M. S. Keane, and J. J. Rousseau (1988). "Extremal Principle Solutions of Games in Characteristic Function Form: Core, Chebychev and Shapley Value Generalizations". In: *Econometrics of Planning and Efficiency*, pp. 123–133.

Covert, I. and S.-I. Lee (2021). "Improving KernelSHAP: Practical Shapley Value Estimation Using Linear Regression". In: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, pp. 3457–3465.

Covert, I. C., C. Kim, and S.-I. Lee (2023). "Learning to Estimate Shapley Values with Vision Transformers". In: *The Eleventh International Conference on Learning Representations*.

Dubey, P., A. Neyman, and R. J. Weber (1981). "Value Theory without Efficiency". *Mathematics of Operations Research*, vol. 6, no. 1, pp. 122–128.

Fumagalli, F., M. Muschalik, P. Kolpaczki, E. Hüllermeier, and B. E. Hammer (2023). "SHAP-IQ: Unified Approximation of any-order Shapley Interactions". In: *Thirty-seventh Conference on Neural Information Processing Systems*.

Ghorbani, A., M. Kim, and J. Zou (2020). "A Distributional Framework for Data Valuation". In: *International Conference on Machine Learning*, pp. 3535–3544.

Ghorbani, A. and J. Zou (2019). "Data Shapley: Equitable Valuation of Data for Machine Learning". In: *International Conference on Machine Learning*, pp. 2242–2251.

Grabisch, M. and C. Labreuche (2001). "How to Improve Acts: An Alternative Representation of the Importance of Criteria in MCDC". *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, no. 02, pp. 145–157.

Harsanyi, J. C. (1963). "A Simplified Bargaining Model for the N-Person Cooperative Game". *International Economic Review*, vol. 4, no. 2, pp. 194–220.

Jethani, N., M. Sudarshan, I. C. Covert, S.-I. Lee, and R. Ranganath (2022). "FastSHAP: Real-Time Shapley Value Estimation". In: *International Conference on Learning Representations*.

Jia, R. et al. (2019). "Towards Efficient Data Valuation Based on the Shapley Value". In: *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1167–1176.

Kingma, D. P. and J. Ba (2014). "Adam: A Method for Stochastic Optimization". In: *ICLR*.

Kolpaczki, P., V. Bengs, and E. Hüllermeier (2023). "Approximating the Shapley Value without Marginal Contributions". arXiv preprint arXiv:2302.00736.

Kwon, Y. and J. Zou (2022a). "Beta Shapley: A Unified and Noise-reduced Data Valuation Framework for Machine Learning". In: *International Conference on Artificial Intelligence and Statistics*, pp. 8780–8802.

Kwon, Y. and J. Y. Zou (2022b). "WeightedSHAP: Analyzing and Improving Shapley Based Feature Attributions". In: *Advances in Neural Information Processing Systems*, pp. 34363–34376.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998). "Gradient-Based Learning Applied to Document Recognition". *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324.

Lin, J., A. Zhang, M. Lécuyer, J. Li, A. Panda, and S. Sen (2022). "Measuring the Effect of Training Data on Deep Learning Predictions via Randomized Experiments". In: *International Conference on Machine Learning*, pp. 13468–13504.

Lundberg, S. M. and S.-I. Lee (2017). "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30*.

Marichal, J.-L. and P. Mathonet (2008). "Approximations of Lovász Extensions and Their Induced Interaction Index". *Discrete Applied Mathematics*, vol. 156, no. 1, pp. 11–24.

Moehle, N., S. Boyd, and A. Ang (2022). "Portfolio Performance Attribution via Shapley Value". *Journal of Investment Management*, vol. 20, no. 3.

Rudin, W. (1953). "Principles of Mathematical Analysis". McGraw-Hill.

Ruiz, L. M., F. Valenciano, and J. M. Zarzuelo (1998). "The Family of Least Square Values for Transferable Utility Games". *Games and Economic Behavior*, vol. 24, no. 1-2, pp. 109–130.

Shapley, L. S. (1953). "A Value for N-Person Games". *Annals of Mathematics Studies*, vol. 28, pp. 307–317.

Wang, J. and R. Jia (2023). "Data Banzhaf: A Robust Data Valuation Framework for Machine Learning". In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*.

Weber, R. J. (1977). "Probabilistic Values for Games". In: *The Shapley Value. Essays in Honor of Lloyd S. Shapley*, pp. 101–119.

Xiao, H., K. Rasul, and R. Vollgraf (2017). "Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms". arXiv preprint arXiv:1708.07747.

Zhang, B., B. Tian, W. Zheng, J. Zhou, and J. Lu (2023a). "Exploring Unified Perspective For Fast Shapley Value Estimation". arXiv preprint arXiv:2311.01010.

Zhang, J., Q. Sun, J. Liu, L. Xiong, J. Pei, and K. Ren (2023b). "Efficient Sampling Approaches to Shapley Value Approximation". In: *ACM SIGMOD International Conference on Management of Data*.

Zhou, Z., X. Xu, R. H. L. Sim, C. S. Foo, and B. K. H. Low (2023). "Probably Approximate Shapley Fairness with Applications in Machine Learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 5, pp. 5910–5918.

Table of Contents

A Distributional Values **12**

B Proofs for the Proposed Algorithms **13**

C Proofs for Convergences **15**

D Least Square Values **19**

E Generalized ARM **20**

F Practical Probabilistic Values **21**

G Practical Aspect of Faster Estimators for Probabilistic Values **21**

H Interpretation of $\kappa(N)$ and $\tau(N)$ in Proposition 4 **23**

I Asymptotic Analysis **24**

J More Experiment Results **26**

A DISTRIBUTIONAL VALUES

Originally, (Ghorbani et al. 2020, Definition 2.2) proposed the Distributional Shapley value defined by

$$\nu(z;\mathcal{D},m,U) = \mathop{\mathbb{E}}_{S\sim\mathcal{D}^{m-1}}[\phi^{Sh}(z;U,S\cup z)]$$

$$\text{where } \phi^{Sh}(z;U,S\cup z) = \sum_{T\subseteq S}\frac{t!(m-1-t)!}{m!}(U(T\cup z)-U(T)).$$

Note that ϕ^{Sh} is just the Shapley value. Then, Ghorbani et al. (2020, Theorem 2.3) proved that

$$\nu(z;\mathcal{D},m,U) = \mathop{\mathbb{E}}_{k\overset{\mathcal{U}}{\sim}[m]}\ \mathop{\mathbb{E}}_{S\sim\mathcal{D}^{k-1}}[U(S\cup z)-U(S)] \tag{10}$$

where $\mathcal{U}$ refers to the uniform distribution. To improve the overall running time, they suggested using a weighted sampling $k\overset{\mathbf{w}}{\sim}[m]$ instead.

However, Eq. (10) holds only with abuse of set operations. Precisely, repeated items are counted in their proof, e.g., $\{x,y,x\}=\{x,x,y\}\neq\{x,y\}$ and $\{z,x,z\}\cup z=\{z,x,z,z\}\neq\{z,x,z\}$. In this work, we stay loyal to the set operations. Regarding Eq. (2), samples from $\mathcal{D}^{s-1}$ are tuples that may contain repeated items, but they are reduced to be sets.

Adhering to the set operations, Eq. (10) is false. Take $\mathcal{D}\leftarrow\frac{1}{2}\delta_x+\frac{1}{2}\delta_y$, $m\leftarrow 2$, $z\leftarrow x$ as example and assume $U(\emptyset)=0$, one can verify that

$$\nu(z;\mathcal{D},m,U) = \frac{3}{4}U(\{x\})+\frac{1}{4}U(\{x,y\})-\frac{1}{4}U(\{y\})$$

$$\mathop{\mathbb{E}}_{k\overset{\mathcal{U}}{\sim}[m]}\ \mathop{\mathbb{E}}_{S\sim\mathcal{D}^{k-1}}[U(S\cup z)-U(S)] = \frac{1}{2}U(\{x\})+\frac{1}{4}U(\{x,y\})-\frac{1}{4}U(\{y\}).$$

Therefore, we refer to Eq. (2) in accordance to the set operations as distributional values.

B PROOFS FOR THE PROPOSED ALGORITHMS

Lemma 1. *Let $\mathbf{J}_n \in \mathbb{R}^{n\times n}$ be the all-one matrix. The matrix $\mathbf{A} = a\mathbf{J}_n + b\mathbf{I}_n$ is invertible if and only if $na + b \neq 0$ and $b \neq 0$. Particularly, $\mathbf{A}^{-1} = -\frac{a}{b(na+b)}\mathbf{J}_n + \frac{1}{b}\mathbf{I}_n$.*

Proof. Observe that the eigenvalues of $\mathbf{J}_n$ are 0 (with $n-1$ independent eigenvectors) and n, and thus the eigenvalues of $a\mathbf{J}_n + b\mathbf{I}_n$ are b and $na + b$. Therefore, $a\mathbf{J}_n + b\mathbf{I}_n$ is invertible if and only if $b \neq 0$ and $na + b \neq 0$.

A priori is that $\mathbf{A}^{-1}$ admits the form of $x\mathbf{J}_n + y\mathbf{I}_n$. Using $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}_n$, we have the equation

$$\begin{gathered}(na+b)x + (a+b)y = 1,\\ (na+b)x + ay = 0,\end{gathered}$$

which leads to that $x = -\frac{a}{b(na+b)}$ and $y = \frac{1}{b}$. □

Proposition 1. *Define, for every $n \geq 1$, $U^n \in \mathcal{G}^n$ and $i \in [n]$,*

$$\mathcal{R}_i(U^n) = \mathcal{R}_i(U^n;\mathcal{P}) = \sum_{\emptyset \subsetneq S \subsetneq [n]} m_s^n \cdot \mathbb{1}_{i\in S} U^n(S) \tag{4}$$

where $m_s^n = m_s^n(\mathcal{P}) = p_s^n + p_{s+1}^n$ for every $s \in [n-1]$. Then, for every $n \geq 1$ and $U^n \in \mathcal{G}^n$, $\mathcal{R}(U^n)$ and $\phi(U^n)$ produce the same ranking. Precisely, $\mathcal{R}(U^n) = \phi(U^n) + g(U^n)\mathbf{1}_n$ where $g(U^n) = g(U^n;\mathcal{P}) \in \mathbb{R}$. As a side note, it holds true for any $\mathbf{p}^n \in \mathbb{R}^n$.

Proof.

$$\begin{aligned}\phi_i(U^n) &= \sum_{S\subseteq [n]\setminus i} p_{s+1}^n (U^n(S\cup i) - U^n(S))\\ &= \sum_{S\subsetneq[n]\colon i\in S} m_s^n U^n(S) + [p_n^n U^n([n]) - \sum_{S\subsetneq [n]} p_{s+1}^n U^n(S)]\\ &= \sum_{\emptyset\subsetneq S\subsetneq[n]} m_s^n \cdot \mathbb{1}_{i\in S} U^n(S) + g(U^n).\end{aligned} \tag{11}$$

□

Proposition 2. *Consider the problem*

$$\operatorname*{arg\,min}_{\mathbf{v}\in\mathbb{R}^n} \sum_{\emptyset\subsetneq S\subsetneq[n]} m_s^n \cdot \left(U^n(S) - \sum_{i\in S} v_i\right)^2, \tag{5}$$

its uniquely optimal solution shares the same ranking as $\phi(U^n)$ for every $n \geq 1$ and $U^n \in \mathcal{G}^n$. As a side note, it holds true for any $\mathbf{p}^n \in \mathbb{R}^n$ that produces non-negative weight vector $\mathbf{m}^n$ with $\sum_{s=1}^{n-1} m_s^n > 0$.

Proof. Let $\mathbf{J}_n \in \mathbb{R}^{n\times n}$ be the all-one matrix, and $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ be the identity matrix. Since the problem (5) is convex, its optimal solution can be obtained by letting its derivative equal 0, which yields

$$\begin{gathered}\mathbf{A}\mathbf{v}^* = \mathbf{b}\\ \text{where } \mathbf{A} = e\mathbf{J}_n + d\mathbf{I}_n \text{ with } e = \sum_{s=2}^{n-1}\binom{n-2}{s-2} m_s^n \text{ and } d = \left(\sum_{s=1}^{n-1}\binom{n-1}{s-1} m_s^n\right) - e,\\ \text{and } \mathbf{b}_i = \sum_{S\subsetneq[n]\colon i\in S} m_s^n \cdot U^n(S) \text{ for every } i\in[n].\end{gathered} \tag{12}$$

Specifically,

$$\begin{aligned}
d &= m_1^n + \sum_{s=2}^{n-1}\left(\binom{n-2}{s-1}+\binom{n-2}{s-2}\right)m_s^n - \sum_{s=2}^{n-1}\binom{n-2}{s-2}m_s^n = \sum_{s=1}^{n-1}\binom{n-2}{s-1}m_s^n \\
&= \sum_{s=1}^{n-1}\binom{n-2}{s-1}p_s^n + \sum_{s=1}^{n-1}\binom{n-2}{s-1}p_{s+1}^n = p_1^n + \sum_{s=2}^{n-1}\binom{n-2}{s-1}p_s^n + \sum_{s=2}^{n-1}\binom{n-2}{s-2}p_s^n + p_n^n \\
&= \sum_{s=1}^{n}\binom{n-1}{s-1}p_s^n = 1.
\end{aligned}$$

By Lemma 1, $\mathbf{A}^{-1} = -\frac{e}{e\cdot n+1}\mathbf{J}_n + \mathbf{I}_n$, which implies the uniqueness of the optimal solution. Therefore,

$$\mathbf{v}^* = \mathbf{A}^{-1}\mathbf{b} = \mathbf{b} + c\mathbf{1}_n \tag{13}$$

where $c = \dfrac{-e}{e\cdot n+1}\sum_{i=1}^n b_i$. According to Eq. (4), we have $\mathbf{b} = \mathcal{R}(U^n)$. Therefore, $\mathbf{v}^*$ and $\phi(U^n)$ share the same ranking. Suppose we have $\mathbf{p}^n \in \mathbb{R}^n$ with $\sum_{s=1}^{n-1} m_s^n > 0$, then there is $\mathbf{v}^* = \frac{1}{d}\mathbf{b} + \hat{c}\mathbf{1}_n$ instead where $\hat{c} = \frac{-e}{e\cdot n+d}\sum_{i=1}^n b_i$. Since $d = \sum_{s=1}^{n-1}\binom{n-2}{s-1}m_s^n > 0$, the factor $\frac{1}{d}$ does not change the ranking of $\mathbf{b}$. ☐

Proposition 3. *The unique optimal solution $\mathbf{v}^*$ to the problem*

$$\operatorname*{arg\,min}_{\mathbf{v}\in\mathbb{R}^{n+1}} \sum_{\emptyset\subsetneq S\subsetneq[n+1]} p_s^n \cdot \left(U^n(S\cap[n]) - \sum_{i\in S} v_i\right)^2 \tag{6}$$

satisfies that $\phi_i(U^n) = v_i^* - v_{n+1}^*$ for every $i \in [n]$.

Proof. For each $U^n \in \mathcal{G}^n$, define $\overline{U}^{n+1} \in \mathcal{G}^{n+1}$ by letting $\overline{U}^{n+1}(S) = U^n(S\cap[n])$. Substituting $\overline{U}^{n+1}(S)$ for $U^n(S\cap[n])$ in the problem (6), and reusing the Eq. (12) accordingly, we have

$$\mathbf{v}^* = \mathbf{A}^{-1}\mathbf{b}$$

$$\text{where } b_i = p_n^n \cdot U^n([n]) + \sum_{S\subsetneq[n]\colon\, i\in S}(p_s^n + p_{s+1}^n)\cdot U^n(S) \quad \forall i\in[n],$$

$$\text{and } b_{n+1} = \sum_{S\subsetneq[n]} p_{s+1}^n \cdot U^n(S).$$

Using Eq. (11), we obtain $\phi_i(U^n) = b_i - b_{n+1}$ for every $i \in [n]$. Meanwhile, by Eq. (13), we have $\mathbf{v}^* = \mathbf{b} + c\mathbf{1}_{n+1}$ for some $c\in\mathbb{R}$, and thus $v_i - v_{n+1} = b_i - b_{n+1} = \phi_i(U^n)$ for every $i\in[n]$. ☐

Theorem 1. *Consider the case when $B = [n]$ (recall that $D_{tr} \equiv [n]$) for the empirical distributional values φ defined in Eq. (7). Let $\mathbf{d}\in\mathbb{R}^m$ be a probability vector that satisfies $d_s \propto \frac{n-s+1}{s}\omega_s$ for every $s\in[m]$, and $\mathbf{v}^*$ be the uniquely optimal solution to*

$$\operatorname*{arg\,min}_{\mathbf{v}\in\mathbb{R}^{n+1}} \ \mathbb{E}_{s\overset{\mathbf{d}}{\sim}[m]}\, \mathbb{E}_{S\overset{\mathcal{U}}{\sim}\overline{\mathcal{B}}_s}\left(U^n(S\cap[n]) - \sum_{i\in S} v_i\right)^2, \tag{8}$$

where $\overline{\mathcal{B}}_s = \{R\subseteq[n+1] \mid |R| = s\}$. There is, for every $i\in[n]$,

$$C\cdot(v_i^* - v_{n+1}^*) = \varphi(i;\mathcal{B},\boldsymbol{\omega},U^n)$$

where $C = \sum_{s=1}^m \frac{n-s+1}{n}\omega_s$. As a side note, $\hat{\mathbf{r}}\in\mathbb{R}^{n+1}$ obtained in Algorithm 2 (substituting $\boldsymbol{\omega}$ for $\mathbf{p}^n$ while letting $\omega_s = 0$ if $s > m$) satisfies that $\mathbb{E}[\hat{r}_i] - \mathbb{E}[\hat{r}_{n+1}] = \varphi(i;\mathcal{B},\boldsymbol{\omega},U^n)$ for every $i\in[n]$.

Proof. Since $U^n(S\cup i) - U^n(S) = 0$ if $i\in S$, for each $i\in[n]$,

$$\begin{aligned}
\varphi(i;\mathcal{B},\boldsymbol{\omega},U^n) &= \sum_{S\subseteq[n]\setminus i\colon\, s<m} \omega_{s+1}\binom{n}{s}^{-1}(U^n(S\cup i) - U^n(S)) \\
&= C\cdot\sum_{S\subseteq[n]\setminus i} q_{s+1}\cdot(U^n(S\cup i) - U^n(S))
\end{aligned}$$

where $q_s = \binom{n}{s-1}^{-1}\frac{\omega_s}{C}$ if $s \in [m]$ and 0 otherwise, and $C = \sum_{s=1}^{m}\binom{n-1}{s-1}\binom{n}{s-1}^{-1}\omega_s = \sum_{s=1}^{m}\frac{n-s+1}{n}\omega_s$. On the other hand, the objective in the problem (8) can be rewritten as

$$\sum_{\emptyset \subsetneq S \subsetneq [n+1]\colon\, s \leq m} d_s \binom{n}{s}^{-1} \left(\overline{U}^{n+1}(S) - \sum_{i \in S} v_i\right)^2$$

where $\overline{U}^{n+1} \in \mathcal{G}^{n+1}$ is defined by letting $\overline{U}^{n+1}(S) = U^n(S \cap [n])$ for every $S \subseteq [n+1]$. Note that $d_s = \alpha\frac{n-s+1}{s}\omega_s$ for every $s \in [m]$ where α is a scalar that makes $\mathbf{d}$ a probability vector. By Proposition 3, for every $i \in [n]$,

$$v_i^* - v_{n+1}^* = \sum_{S \subseteq [n]\backslash i} \hat{q}_{s+1} \cdot (U^n(S \cup i) - U^n(S))$$

where $\hat{q}_s = \beta \cdot d_s\binom{n}{s}^{-1}$ for every $s \in [m]$ and 0 otherwise. Suffice it to show that $\hat{q}_s = q_s$ for every $s \in [m]$. Note that

$$1 = \sum_{s=1}^{m}\binom{n-1}{s-1}\hat{q}_s = \sum_{s=1}^{m}\binom{n-1}{s-1}\beta \cdot \alpha\frac{n-s+1}{s}\omega_s\binom{n}{s}^{-1} = \alpha \cdot \beta \cdot C,$$

and thus

$$q_s = \beta \cdot \alpha\binom{n}{s-1}^{-1}\omega_s = \beta\binom{n}{s-1}^{-1}\frac{s}{n-s+1}d_s = \beta\binom{n}{s}^{-1}d_s = \hat{q}_s.$$

For the side note, note that

$$L \cdot (\mathbb{E}[\hat{r}_i] - \mathbb{E}[\hat{r}_{n+1}]) = v_i^* - v_{n+1}^*$$

where

$$L = \sum_{s=1}^{n}\frac{s}{n+1}\binom{n+1}{s}q_s = \sum_{s=1}^{m}\frac{s}{n+1}\binom{n+1}{s}\binom{n}{s-1}^{-1}\frac{\omega_s}{C} = \frac{1}{C}.$$

□

C PROOFS FOR CONVERGENCES

Proposition 5. *Assume that $\|U^n\|_\infty \leq u$ for every $U^n \in \bigcup_{n\geq 1}\mathcal{G}^n$, any estimator based on the sampling lift strategy requires $O(\frac{n^2}{\epsilon^2}\log\frac{n}{\delta})$ utility evaluations to achieve $P(\|\hat{\phi}(U^n) - \phi(U^n)\|_2 \geq \epsilon) \leq \delta$.*

The sampling lift strategy refers to any estimator designed according to

$$\phi_i(U^n) = \sum_{S \subseteq [n]\backslash i} p_{s+1}^n\,(U^n(S \cup i) - U^n(S)) = \mathbb{E}_S[U^n(S \cup i) - U^n(S)]$$

for every $i \in [n]$ as $\sum_{s=1}^{n}\binom{n-1}{s-1}p_s^n = 1$. Precisely, $U^n(S \cup i) - U^n(S)$ is a random variable following a certain probability distribution, denoted by P_i, on all subsets $S \subseteq [n]\backslash i$. Let T be the total number of samples and $\mathcal{S} = \{(S_1^1, S_1^2, \ldots, S_1^n), (S_2^1, S_2^2, \ldots, S_2^n), \ldots, (S_T^1, S_T^2, \ldots, S_T^n)\}$ be a list of T samples where each $\mathcal{S}^i = \{S_1^i, S_2^i, \ldots, S_T^i\}$ are sampled from P_i. Then, the estimator based on the sampling lift strategy is $\hat{\phi}_i(U^n) = \frac{1}{T}\sum_{k=1}^{T}(U^n(S_k^i \cup i) - U^n(S_k^i))$. There are two slightly different implementations: i) sampling each $\mathcal{S}^i$ independently, and ii) sampling $\mathcal{S}^i$ for some $i \in [n]$, and then reusing all samples in $\mathcal{S}^i$ for every other $j \in [n]$ by swapping i and j. Nevertheless, the proof below works for both of them. Besides, the proof is adapted from (Wang and Jia 2023, Theorem 4.8) where they stated for the Banzhaf value. Nevertheless, we point out that it applies to all probabilistic values.

Proof. For simplicity, we write $\boldsymbol{\phi} = \phi(U^n)$ and $\hat{\boldsymbol{\phi}} = \hat{\phi}(U^n)$. Fix an $i \in [n]$, since $\mathbb{E}[\hat{\phi}_i] = \phi_i$, by the Hoeffding's inequality,

$$P(|\hat{\phi}_i - \phi_i| \geq \epsilon) \leq 2\exp(-\frac{T\epsilon^2}{2u^2}).$$

Then,

$$P(\|\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}\|_2 \geq \epsilon) \leq P(\bigcup_{i\in[n]} |\hat{\phi}_i - \phi_i| \geq \frac{\epsilon}{\sqrt{n}}) \leq 2n \exp(-\frac{T\epsilon^2}{2nu^2})$$

Letting $\delta \geq 2n\exp(-\frac{T\epsilon^2}{2nu^2})$ leads to $T \geq \frac{2nu^2}{\epsilon^2}\log\frac{2n}{\delta}$. Since each sample $(S_k^1, S_k^2, \ldots, S_k^n)$ invokes $2n$ utility evaluations, it requires $O(\frac{n^2}{\epsilon^2}\log\frac{n}{\delta})$ utility evaluations to achieve an (ϵ, δ)-approximation. □

Proposition 4. *Assume that* $\|U^n\|_\infty \leq u$ *for every* $U^n \in \bigcup_{n\geq 1}\mathcal{G}^n$. *GELS and GELS-Shapley estimators require* $T = O(\frac{\tau(n)n}{\epsilon^2}\log\frac{n}{\delta})$ *samples, which is equal to the number of utility evaluations, to achieve* $P(\|\hat{\phi}(U^n) - \phi(U^n)\| \geq \epsilon) \leq \delta$. *For GELS-R estimator, it requires* $T = O(\frac{\kappa(n)n}{\epsilon^2}\log\frac{n}{\delta})$ *samples (equivalently, utility evaluations) instead to achieve an* (ϵ, δ)*-approximation.*

Proof. The proof is adapted from (Wang and Jia 2023, Theorem 4.9). First, we prove the convergence for GELS-R. Note that it is an unbiased estimator of

$$\mathbf{r} = \left(\sum_{s=1}^{n-1}\binom{n-1}{s-1}(p_s^n + p_{s+1}^n)\right)^{-1}\mathcal{R}(U^n). \tag{14}$$

Let $\gamma(n) = \frac{\sum_{s=1}^{n-1}\binom{n-1}{s-1}(p_s^n + p_{s+1}^n)}{\sum_{s=1}^{n-1}\binom{n}{s}(p_s^n + p_{s+1}^n)}$. For convenience, we write $\mathcal{S} = \{S_1, S_2, \ldots, S_T\}$ that contains all sampled subsets, and $T_i = |\{S \in \mathcal{S} \mid i \in S\}|$ for every $i \in [n]$. For every $i \in [n]$, define

$$\overline{r}_i = \frac{1}{\gamma(n)T}\sum_{S\in\mathcal{S}\colon i\in S} U^n(S).$$

Then, we have

$$|\hat{r}_i - \overline{r}_i| = \left|\left(\frac{1}{T_i} - \frac{1}{\gamma(n)T}\right)\sum_{S\in\mathcal{S}\colon i\in S} U^n(S)\right| \leq \frac{u}{\gamma(n)T}|\gamma(n)T - T_i|.$$

Note that this inequality $|\hat{r}_i - \overline{r}_i| \leq \dfrac{u}{\gamma(n)T}|\gamma(n)T - T_i|$ still holds when $T_i = 0$. Since $T_i \sim \mathrm{binomial}(T, \gamma(n))$, by the Hoeffding's inequality, there is

$$P(|T_i - \gamma(n)T| \geq \Delta) \leq 2\exp(-\frac{2\Delta^2}{T}).$$

Therefore, $|\hat{r}_i - \overline{r}_i| < \dfrac{u\Delta}{\gamma(n)T}$ provided that $|T_i - \gamma(n)T| < \Delta$. Since $\overline{r}_i = \frac{1}{T}\sum_{S\in\mathcal{S}}\beta(s,n)U^n(S)$ where $\beta(s,n) = \gamma(n)^{-1}$ if $i \in S$ and 0 otherwise, and $\mathbb{E}[\overline{r}_i] = r_i$, using the Heoffding's inequality again yields

$$P(|\overline{r}_i - r_i| \geq \sigma) \leq 2\exp(-\frac{\gamma(n)^2 T\sigma^2}{2u^2})$$

Therefore,

$$\begin{aligned} P(|\hat{r}_i - r_i| \geq \epsilon) &= P(|\hat{r}_i - r_i| \geq \epsilon \cap |T_i - \gamma(n)T| < \Delta) + P(|\hat{r}_i - r_i| \geq \epsilon \cap |T_i - \gamma(n)T| \geq \Delta) \\ &\leq P(|\hat{r}_i - r_i| \geq \epsilon \mid |T_i - \gamma(n)T| < \Delta) + 2\exp(-\frac{2\Delta^2}{T}) \\ &\leq P(|\overline{r}_i - r_i| \geq \epsilon - \frac{u\Delta}{\gamma(n)T} \mid |T_i - \gamma(n)T| < \Delta) + 2\exp(-\frac{2\Delta^2}{T}) \\ &\leq \frac{P(|\overline{r}_i - r_i| \geq \epsilon - \frac{u\Delta}{\gamma(n)T})}{1 - 2\exp(-\frac{2\Delta^2}{T})} + 2\exp(-\frac{2\Delta^2}{T}) \\ &\leq \frac{2\exp(-\frac{\gamma(n)^2T\left(\epsilon - \frac{u\Delta}{\gamma(n)T}\right)^2}{2u^2})}{1 - 2\exp(-\frac{2\Delta^2}{T})} + 2\exp(-\frac{2\Delta^2}{T}) \\ &\leq 3\exp(-\frac{\gamma(n)^2T\left(\epsilon - \frac{u\Delta}{\gamma(n)T}\right)^2}{2u^2}) + 2\exp(-\frac{2\Delta^2}{T}) \end{aligned}$$

where $1 - 2\exp(-\frac{2\Delta^2}{T}) \geq \frac{2}{3}$ provided that T is sufficiently large. The next step is to determine Δ by solving the equation $-\frac{\gamma(n)^2T\left(\epsilon - \frac{u\Delta}{\gamma(n)T}\right)^2}{2u^2} = -\frac{2\Delta^2}{T}$, which yields $\Delta = \frac{\gamma(n)T\epsilon}{3u}$. Note that this solution gives $\epsilon - \frac{u\Delta}{\gamma(n)T} = \frac{2\epsilon}{3} > 0$ and $\frac{2\Delta^2}{T} = \frac{2\gamma(n)^2T\epsilon^2}{9u^2}$. Besides, the inequality $1 - 2\exp(-\frac{2\Delta^2}{T}) \geq \frac{2}{3}$ leads to $T \geq \frac{9\log(6)u^2}{2\gamma(n)^2\epsilon^2}$. Eventually, we have

$$P(|\hat{r}_i - r_i| \geq \epsilon) \leq 5\exp(-\frac{2\gamma(n)^2T\epsilon^2}{9u^2}) \tag{15}$$

provided that $T \geq \frac{9\log(6)u^2}{2\gamma(n)^2\epsilon^2}$. Then,

$$P(\|\hat{\mathbf{r}} - \mathbf{r}\|_2 \geq \epsilon) \leq P(\bigcup_{i\in[n]} |\hat{r}_i - r_i| \geq \frac{\epsilon}{\sqrt{n}}) \leq 5n\exp(-\frac{2\gamma(n)^2T\epsilon^2}{9u^2n}). \tag{16}$$

Solving $5n\exp(-\frac{2\gamma(n)^2T\epsilon^2}{9u^2n}) \leq \delta$ leads to $T \geq \frac{9u^2n}{2\gamma(n)^2\epsilon^2}\log\frac{5n}{\delta}$, and thus GELS-R requires

$$\max(\frac{9\log(6)u^2}{2\gamma(n)^2\epsilon^2}, \frac{9u^2n}{2\gamma(n)^2\epsilon^2}\log\frac{5n}{\delta}) = O(\frac{\kappa(n)n}{\epsilon^2}\log\frac{n}{\delta})$$

utility evaluations to achieve an (ϵ, δ)-approximation where $\kappa(n) = \gamma(n)^{-2}$.

Next, we prove the convergence for GELS. For simplicity, we write $\boldsymbol{\phi} = \phi(U^n)$ and $\mathbf{h} = \left(\sum_{s=1}^{n}\binom{n}{s-1}p_s^n\right)^{-1}\mathcal{R}(\overline{U}^{n+1})$, see Eq. (14). By Eq. (15), there is, for every $i \in [n+1]$,

$$P(|\hat{h}_i - h_i| \geq \epsilon) \leq 5\exp(-\frac{2\widetilde{\gamma}(n)^2T\epsilon^2}{9u^2})$$

provided that $T \geq \frac{9\log(6)u^2}{2\widetilde{\gamma}(n)^2\epsilon^2}$ where $\widetilde{\gamma}(n) = \frac{\sum_{s=1}^{n}\binom{n}{s-1}p_s^n}{\sum_{s=1}^{n}\binom{n+1}{s}p_s^n}$. Therefore, for every $i \in [n]$,

$$\begin{aligned} P(|(\hat{h}_i - \hat{h}_{n+1}) - (h_i - h_{n+1})| \geq \epsilon) &\leq P(|\hat{h}_i - h_i| \geq \frac{\epsilon}{2} \cup |\hat{h}_{n+1} - h_{n+1}| \geq \frac{\epsilon}{2}) \\ &\leq 10\exp(-\frac{\widetilde{\gamma}(n)^2T\epsilon^2}{18u^2}). \end{aligned}$$

let $\chi(n) = \sum_{s=1}^{n}\binom{n}{s-1}p_s^n$ and $\eta(n) = \sum_{s=1}^{n}\binom{n+1}{s}p_s^n$, and note that $\widetilde{\gamma}(n) = \frac{\chi(n)}{\eta(n)}$. As argued in Remark 1, $\mathcal{R}_i(\overline{U}^{n+1}) - \mathcal{R}_{n+1}(\overline{U}^{n+1}) = \phi_i(U^n)$ for every $i \in [n]$, and thus $\chi(n)(h_i - h_{n+1}) =$

$\phi_i(U^n)$ for every $i \in [n]$. So, for every $i \in [n]$,

$$P(|\hat{\phi}_i - \phi_i| \geq \epsilon) = P(|(\hat{h}_i - \hat{h}_{n+1}) - (h_i - h_{n+1})| \geq \frac{\epsilon}{\chi(n)}) \leq 10\exp(-\frac{T\epsilon^2}{18u^2\eta(n)^2}).$$

Therefore,

$$P(\|\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}\|_2 \geq \epsilon) \leq P(\bigcup_{i\in[n]} |\hat{\phi}_i - \phi_i| \geq \frac{\epsilon}{\sqrt{n}}) \leq 10n\exp(-\frac{T\epsilon^2}{18u^2n\eta(n)^2}).$$

Solving $\delta \geq 10n\exp(-\frac{T\epsilon^2}{18u^2n\eta(n)^2})$ yields $T \geq \frac{18u^2n\eta(n)^2}{\epsilon^2}\log\frac{10n}{\delta}$. To conclude, GELS requires

$$\max(\frac{9\log(6)u^2}{2\widetilde{\gamma}(n)^2\epsilon^2}, \frac{18u^2n\eta(n)^2}{\epsilon^2}\log\frac{10n}{\delta}) = O(\frac{\tau(n)n}{\epsilon^2}\log\frac{n}{\delta})$$

utility evaluations to achieve an (ϵ, δ)-approximation where $\tau(n) = \eta(n)^2$. Note that $\widetilde{\gamma}(n)^{-1} \leq \eta(n)$ as $\chi(n) = \sum_{s=1}^{n}\binom{n}{s-1}\binom{n-1}{s-1}^{-1}\binom{n-1}{s-1}p_s^n = \sum_{s=1}^{n}\frac{n}{n-s+1}\binom{n-1}{s-1}p_s^n \geq 1$.

Last, we prove the convergence for GELS-Shapley. Reusing Eq. (14), since $p_s^n = \frac{(s-1)!(n-s)!}{n!}$ for the Shapley value, it is $H_{n-1}\cdot\mathbf{r} = \mathcal{R}(U^n)$ where $H_{n-1} = \sum_{s=1}^{n-1}\binom{n-1}{s-1}(p_s^n + p_{s+1}^n) = \sum_{s=1}^{n-1}\frac{1}{s}$. By Eq. (16), there is

$$P(H_{n-1}\cdot\|\hat{\mathbf{r}} - \mathbf{r}\|_2 \geq \epsilon) \leq 5n\exp\left(-\frac{2T\epsilon^2}{9u^2n\eta(n-1)^2}\right)$$

provided that $T \geq \frac{9\log(6)u^2}{2\gamma(n)^2\epsilon^2}$. Note that $\sum_{s=1}^{n-1}\binom{n}{s}(p_s^n + p_{s+1}^n) = \sum_{s=1}^{n-1}\binom{n}{s}p_s^{n-1} = \eta(n-1)$.

For convenience, let $\boldsymbol{\phi}^{Sh}$ be the corresponding Shapley value. As discussed in Remark 2,

$$\boldsymbol{\phi}^{Sh} = H_{n-1}\cdot\mathbf{r} + \frac{U^n([n]) - U^n(\emptyset) - \sum_{s=1}^{n} H_{n-1}\cdot r_i}{n}.$$

The procedure of casting $H_{n-1}\cdot\mathbf{r}$ into $\boldsymbol{\phi}^{Sh}$ is called additive-efficient-normalization (Ruiz et al. 1998, Definition 11). Particularly, we write

$$\hat{\boldsymbol{\phi}}^{Sh} = H_{n-1}\cdot\hat{\mathbf{r}} + \frac{U^n([n]) - U^n(\emptyset) - \sum_{s=1}^{n} H_{n-1}\cdot\hat{r}_i}{n}.$$

As pointed out by Jethani et al. (2022, Appendix B), the additive-efficient-normalization is just an orthogonal projection, and therefore we have

$$\|\hat{\boldsymbol{\phi}}^{Sh} - \boldsymbol{\phi}^{Sh}\|_2 \leq \|H_{n-1}\cdot\hat{\mathbf{r}} - H_{n-1}\cdot\mathbf{r}\,\|_2,$$

which suggest

$$P(\|\hat{\boldsymbol{\phi}}^{Sh} - \boldsymbol{\phi}^{Sh}\|_2 \geq \epsilon) \leq P(\|H_{n-1}\cdot\hat{\mathbf{r}} - H_{n-1}\cdot\mathbf{r}\,\|_2 \geq \epsilon) \leq 5n\exp\left(-\frac{2T\epsilon^2}{9u^2n\eta(n-1)^2}\right).$$

Solving $\delta \geq 5n\exp\left(-\frac{2T\epsilon^2}{9u^2n\eta(n-1)^2}\right)$ yields $T \geq \frac{9u^2n\eta(n-1)^2}{2\epsilon^2}\log\frac{5n}{\delta}$. Since $\gamma(n)^{-1} \leq \eta(n-1)$, we eventually have that GELS-Shapley requires $O(\frac{n\tau(n)}{\epsilon^2}\log\frac{n}{\delta})$ utility evaluations to achieve an (ϵ, δ)-approximation where $\tau(n) = \eta(n)^2 > \eta(n-1)^2$. Note that i) for the Shapley value, $\eta(n) = \sum_{s=1}^{n}\binom{n+1}{s}\binom{n-1}{s-1}^{-1}\frac{1}{n} = \sum_{s=1}^{n}\frac{n+1}{s(n+1-s)} = \sum_{s=1}^{n}\left(\frac{1}{s} + \frac{1}{n+1-s}\right) = 2H_n$, and thus $\eta(n-1) < \eta(n)$; and ii) $\gamma(n)^{-1} \leq \eta(n-1)$ as $\gamma(n)\cdot\eta(n-1) = H_{n-1} \geq 1$. $\square$

D LEAST SQUARE VALUES

In this appendix, we demonstrate how we obtained proposition 2 at the very beginning without knowing proposition 1.

Definition 1 (Least Square Values (Ruiz et al. 1998, Definition 5))**.** *Suppose a non-negative non-zero vector* $\mathbf{m} \in \mathbb{R}^{n-1}$ *is given, a least square value* $\xi(U^n; \mathbf{m}) \in \mathbb{R}^n$ *is defined to be the uniquely optimal solution to the problem*

$$\begin{aligned} &\arg\min_{\mathbf{v}\in\mathbb{R}^n} \sum_{\emptyset\subsetneq S\subsetneq[n]} m_s \left(U^n(S) - U^n(\emptyset) - \sum_{i\in S} v_i\right)^2 \\ &\textit{s.t.} \sum_{i=1}^{n} v_i = U^n([n]) - U^n(\emptyset). \end{aligned} \tag{17}$$

Specifically, Ruiz et al. (1998, Theorem 8) developed a system of axioms that uniquely characterizes the family of least square values. Moreover, its relationship with the family of probabilistic values was also revealed.

Proposition 6 (Ruiz et al. 1998, Theorem 12)**.** *For each vector of weights* $\mathbf{p}^n \in \mathbb{R}^n$ *that satisfies* $\sum_{s=1}^{n}\binom{n-1}{s-1}p_s^n = 1$*, define* $\mathbf{m} \in \mathbb{R}^{n-1}$ *by letting* $m_s = p_s^n + p_{s+1}^n$*, there exists some function* $f : \mathcal{G}^n \to \mathbb{R}$ *such that*

$$\xi_i(U^n; \mathbf{m}) = \sum_{S\subseteq[n]\backslash i} p_s^n \cdot (U^n(S\cup i) - U^n(\emptyset)) + f(U^n)$$

for every $U^n \in \mathcal{G}^n$ *and* $i \in [n]$*.*

As proposed by Ruiz et al. (1998, Definition 11), for each probabilistic value $\phi(U^n)$, its additive-efficient-normalization $\overline{\phi}$ is defined to be

$$\overline{\phi}(U^n) = \phi(U^n) + \frac{1}{n}(U^n([n]) - U^n(\emptyset) - \sum_{i\in[n]} \phi_i(U^n)).$$

Using Proposition 6, one can verify that $\xi(U^n; \mathbf{m}) = \overline{\phi}(U^n)$ and $f(U^n) = \frac{1}{n}(U^n([n]) - U^n(\emptyset) - \sum_{i\in[n]} \phi_i(U^n))$. On the other hand, we notice that the constraint of the problem (17) and the term $-U^n(\emptyset)$ in the objective can be removed if it is the ranking that matters.

Proposition 7. *Suppose a utility function* $U^n \in \mathcal{G}^n$ *and a non-negative non-zero vector* $\mathbf{m} \in \mathbb{R}^{n-1}$ *are given, the uniquely optimal solution* $\mathbf{u}^*$ *to the problem*

$$\arg\min_{\mathbf{u}\in\mathbb{R}^n} \sum_{\emptyset\subsetneq S\subsetneq[n]} m_s \left(U^n(S) - \sum_{i\in S} u_i\right)^2 \tag{18}$$

satisfies that there exists some $c \in \mathbb{R}$ *such that*

$$\mathbf{u}^* = \xi(U^n; \mathbf{m}) + c\mathbf{1}_n.$$

Proof. Since the optimization problem (17) is convex, it can be solved using the KKT condition. Specifically, by introducing a dual variable $\lambda \in \mathbb{R}$, there is

$$\mathbf{v}^\top \mathbf{A}\mathbf{v} - 2\mathbf{b}^\top \mathbf{v} + \sum_{\emptyset\subsetneq S\subsetneq[n]} m_s \left(U^n(S) - U^n(\emptyset)\right)^2 - 2\lambda(U^n([n]) - U^n(\emptyset) - \sum_{i=1}^{n} \phi_i)$$

$$\text{where } A_{ij} = \begin{cases} \sum_{s=1}^{n-1}\binom{n-1}{s-1} m_s, & i = j \\ \sum_{s=2}^{n-1}\binom{n-2}{s-2} m_s, & i \neq j \end{cases}, \text{ and } b_i = \sum_{S\subsetneq[n]\colon i\in S} m_s \left(U^n(S) - U^n(\emptyset)\right) \quad \forall i \in [n].$$

Letting its derivative equal 0 yields

$$\mathbf{A}\mathbf{v} - \mathbf{b} + \lambda\mathbf{1}_n = 0 \text{ and } \mathbf{1}_n^\top \mathbf{v} = U^n([n]) - U^n(\emptyset),$$

which leads to the uniquely optimal solution

$$\mathbf{v}^* = \mathbf{A}^{-1}\left(\mathbf{b} - \frac{\mathbf{1}_n^\top \mathbf{A}^{-1}\mathbf{b} - U^n([n]) + U^n(\emptyset)}{\mathbf{1}_n^\top \mathbf{A}^{-1}\mathbf{1}_n}\mathbf{1}_n\right). \tag{19}$$

Specifically,

$$\begin{aligned}&\sum_{s=1}^{n-1}\binom{n-1}{s-1}m_s - \sum_{s=2}^{n-1}\binom{n-2}{s-2}m_s\\ =&m_1 + \sum_{s=2}^{n-1}\left(\binom{n-2}{s-1}+\binom{n-2}{s-2}\right)m_s - \sum_{s=2}^{n-1}\binom{n-2}{s-2}m_s = \sum_{s=1}^{n-1}\binom{n-2}{s-1}m_s > 0,\end{aligned}$$

by lemma 1, the existence of $\mathbf{A}^{-1}$ is guaranteed. On the other hand, letting the derivative of the objective in the problem (18) equal 0 gives

$$\mathbf{u}^* = \mathbf{A}^{-1}(\mathbf{b} + t\mathbf{1}_n) \text{ where } t = U^n(\emptyset)\cdot A_{11}.$$

Therefore, we eventually have, for some $c \in \mathbb{R}$,

$$\mathbf{u}^* = \mathbf{v}^* + c\mathbf{1}_n = \xi(U^n; \mathbf{m}) + c\mathbf{1}.$$

for some $c \in \mathbb{R}$. □

Remark 5. *To conclude, Proposition 2 is derived from Propositions 6 and 7.*

E GENERALIZED ARM

As proposed by Kolpaczki et al. (2023), the Shapley value can be rewritten as, for every $U^n \in \mathcal{G}^n$ and $i \in [n]$,

$$\phi_i^{Sh}(U^n) = \mathbb{E}_{S\sim P_{ARM}^+ \mid i\in S}[U^n(S)] - \mathbb{E}_{S\sim P_{ARM}^- \mid i\notin S}[U^n(S)]$$

where $P_{ARM}^+(S) = \frac{1}{s\cdot H}\binom{n}{s}^{-1}$ for every $S \in \mathcal{S}^+ = \{\emptyset \subsetneq S \subseteq [n]\}$ and $P_{ARM}^-(S) = \frac{1}{(n-s)\cdot H}\binom{n}{s}^{-1}$ for every $S \in \mathcal{S}^- = \{S \subsetneq [n]\}$; $H = \sum_{s=1}^{n}\frac{1}{s}$.

The estimator based on this formula is called the approximation-without-requiring-marginal estimator. We found that this methodology can be easily adapted for every other probabilistic value, which is summarized in the below.

Proposition 8. *For every $U^n \in \mathcal{G}^n$ and $i \in [n]$,*

$$\phi_i(U^n) = \mathbb{E}_{S\sim P_{ARM}^+ \mid i\in S}[U^n(S)] - \mathbb{E}_{S\sim P_{ARM}^- \mid i\notin S}[U^n(S)]$$

where $P_{ARM}^+(S) \propto p_s$ for every $S \in \mathcal{S}^+$ and $P_{ARM}^-(S) \propto p_{s+1}$ for every $S \in \mathcal{S}^-$.

Proof. For simplicity, we write $\mathbf{p}^+$ and $\mathbf{p}^-$ instead of P_{ARM}^+ and P_{ARM}^-. Precisely, $p_s^+ = \alpha\cdot p_s$ and $p_s^- = \beta\cdot p_{s+1}$ such that $\sum_{s=1}^{n}\binom{n}{s}p_s^+ = 1$ and $\sum_{s=0}^{n-1}\binom{n}{s}p_{s+1}^- = 1$. Let S^+ and S^- be the corresponding random samples from $\mathcal{S}^+$ and $\mathcal{S}^-$, respectively. Fix an $i \in [n]$, observe that,

$$\begin{aligned}P(i\in S^+) &= \sum_{s=1}^{n} P(i\in S^+ \mid |S^+| = s)\cdot P(|S^+| = s) = \sum_{s=1}^{n}\binom{n-1}{s-1}\binom{n}{s}^{-1}\cdot\binom{n}{s}p_s^+\\ &= \alpha\sum_{s=1}^{n}\binom{n-1}{s-1}p_s = \alpha.\end{aligned}$$

Therefore,

$$P(S^+ \mid i \in S^+) = \frac{P(S^+, i\in S^+)}{P(i\in S^+)} = \frac{\alpha\cdot p_s}{\alpha} = p_s.$$

Similarly,

$$P(i \notin S^-) = \sum_{s=0}^{N-1} P(i \notin S^- \mid |S^-| = s) \cdot P(|S^-| = s) = \sum_{s=0}^{n-1} \binom{n-1}{s} \binom{n}{s}^{-1} \cdot \binom{n}{s} p_s^-$$

$$= \beta \sum_{s=0}^{n-1} \binom{n-1}{s} p_{s+1} = \beta,$$

which leads to

$$P(S^- \mid i \notin S^-) = \frac{P(S^-, i \notin S^-)}{P(i \notin S^-)} = \frac{\beta \cdot p_{s+1}}{\beta} = p_{s+1}.$$

$\square$

F PRACTICAL PROBABILISTIC VALUES

As provided by Dubey et al. (1981), each semi-value, which is a subfamily of probabilistic values, can be expressed as, for every $U^n \in \mathcal{G}^n$ and $i \in [n]$,

$$\phi_i(U^n) = \phi_i(U^n; \mu) = \sum_{S \subseteq [n] \setminus i} \left(\int_0^1 t^s (1-t)^{n-1-s} \mathrm{d}\mu(t) \right) (U^n(S \cup i) - U^n(S))$$

where μ is any probability measure on the closed interval $[0, 1]$. In other words, in view of Eq. (1), there is $p_s^n = \int_0^1 t^{s-1}(1-t)^{n-s} \mathrm{d}\mu(t)$.

To the best of our knowledge, practical semi-values, i.e., those ever studied in the previous references, include the Banzhaf value (Wang and Jia 2023) and the Beta Shapley values with $\alpha, \beta \geq 1$ (Kwon and Zou 2022a; Kwon and Zou 2022b). Note that Beta$(1, 1)$ is exactly the Shapley value. For the Banzhaf value, the corresponding μ is the Dirac delta distribution $\delta_{0.5}$, which leads to $p_s^n = 2^{-(n-1)}$. For Beta(α, β), the corresponding probability density function for μ is $\propto t^{\beta-1}(1-t)^{\alpha-1}$, and thus $p_s^n = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \cdot \frac{\Gamma(\beta+s-1)\Gamma(\alpha+n-s)}{\Gamma(\alpha+\beta+n-1)}$. Specifically, it is $p_s^n = \frac{(s-1)!(n-s)!}{n!}$ for Beta$(1, 1)$, i.e., the Shapley value.

G PRACTICAL ASPECT OF FASTER ESTIMATORS FOR PROBABILISTIC VALUES

As far as we know, in feature attribution, FastSHAP is the only framework of training semi-value-based explainers (Jethani et al. 2022), which is based on the least square regression Eq. (3) specific for the Shapley value. Recently, Kwon and Zou (2022b) showed that other candidates of the Beta Shapley values tend to perform better than the Shapley value in feature attribution. Therefore, one may ask how to cast other probabilistic values into optimization, which is answered by Propositions 1 and 3. Though AME proposed by Lin et al. (2022) provides an alternative way to achieve this goal, it is restricted to a subfamily of semi-values, which does not include, e.g., the Shapley value and Beta$(4, 1)$ used in our experiments. Besides, as shown in our experiments of comparing convergences, on Beta$(2, 2)$, the induced estimator of AME does not rival our GELS-R and GELS estimators, which are derived from solving the least square regressions (5) and (6).

Recall that the optimization problem used by FastSHAP is

$$\mathbb{E}_{z \in \mathcal{Z}} \left[\sum_{\emptyset \subsetneq S \subsetneq [d]} \frac{d-1}{\binom{d}{s} s(d-s)} \left(U_z(S) - U_z(\emptyset) - \mathbf{1}_S^\top \phi(z; \boldsymbol{\theta}) \right)^2 \right]$$

where d is the number of features (a counterpart of n), $\phi(z; \boldsymbol{\theta}) \in \mathbb{R}^d$ is a trainable explainer parameterized by $\boldsymbol{\theta}$, U_z is a utility function based on the data point $z = (\mathbf{x}, y)$ where $\mathbf{x}$ and y represent the features and label, respectively, and $\mathbf{1}_S \in \mathbb{R}^d$ is defined by $\mathbf{1}_S(i) = 1$ if $i \in S$ and 0 otherwise.

Assume $\{S\}_{\emptyset \subsetneq S \subsetneq [d]}$ is ordered so that each utility function U_z can be treated as a vector $\mathbf{U}_z \in \mathbb{R}^{2^d-2}$. Besides, let $\mathbf{W} \in \mathbb{R}^{(2^d-2)\times(2^d-2)}$ be a diagonal matrix such that $W(S, S) = \frac{d-1}{\binom{d}{s} s(d-s)}$, and define $\mathbf{X} \in \mathbb{R}^{(2^d-2)\times d}$ by letting the S-th row of $\mathbf{X}$ is $\mathbf{1}_S$. Very recently, Zhang et al. (2023a) discovered that

$$\begin{aligned}
&\mathbb{E}_{z\in\mathcal{Z}}\left[\|\phi(z;\boldsymbol{\theta}) - (\mathbf{X}^\top\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{W}\left(\mathbf{U}_z - U_z(\emptyset)\mathbf{1}_{2^d-2}\right)\|^2_{\mathbf{X}^\top\mathbf{W}\mathbf{X}}\right] \\
&\quad= \mathbb{E}_{z\in\mathcal{Z}}\left[\|\mathbf{U}_z - U_z(\emptyset)\mathbf{1}_{2^d-2} - \mathbf{X}\phi(z;\boldsymbol{\theta})\|^2_{\mathbf{W}} + C_z\right] \\
&\quad= \mathbb{E}_{z\in\mathcal{Z}}\left[\sum_{\emptyset\subsetneq S\subsetneq[d]} \frac{d-1}{\binom{d}{s}s(d-s)}\left(U_z(S) - U_z(\emptyset) - \mathbf{1}_S^\top\phi(z;\boldsymbol{\theta})\right)^2 + C_z\right]
\end{aligned}$$

where

$$C_z = \|(\mathbf{X}^\top\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{W}(\mathbf{U}_z - U_z(\emptyset)\mathbf{1}_{2^d-2})\|^2_{\mathbf{X}^\top\mathbf{W}\mathbf{X}} - \|U_z(\emptyset)\mathbf{1}_{2^d-2} - \mathbf{U}_z\|^2_{\mathbf{W}}.$$

The convention is $\|\mathbf{z}\|^2_{\mathbf{A}} \coloneqq \mathbf{z}^\top\mathbf{A}\mathbf{z}$. This relationship reveals that FastSHAP is just to train explainers towards a (potentially) biased target $(\mathbf{X}^\top\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{W}\left(\mathbf{U}_z - U_z(\emptyset)\mathbf{1}_{2^d-2}\right)$ under the metric induced by $\mathbf{X}^\top\mathbf{W}\mathbf{X}$; see Proposition 9. Therefore, they instead proposed to train explainers based on

$$\mathbb{E}_{z\sim\mathcal{Z}}[\|\phi(z;\theta) - \phi\|^2] \tag{20}$$

where ϕ is any unbiased estimator (though they employed the one they discovered) for the Shapley value. They showed in the experiments that this framework of training explainers is not just simple but effective. Note that $(\mathbf{X}^\top\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{W}\left(\mathbf{U}_z - U_z(\emptyset)\mathbf{1}_{2^d-2}\right)$ is expected to be biased because compared with Eq. (3) the efficiency constraint, which is supposed to be $\sum_{i=1}^d \phi(z;\boldsymbol{\theta}) = U_z([d]) - U_z(\emptyset)$, has been removed. Nevertheless, to overcome this issue, the authors of FastSHAP added an additive-efficient-normalization layer on top during training and inference (Jethani et al. 2022, Table 3), which is $\phi(z;\boldsymbol{\theta}) \leftarrow \phi(z;\boldsymbol{\theta}) + \frac{1}{d}(U_z([d]) - U_z(\emptyset) - \sum_{i=1}^d \phi_i(z;\boldsymbol{\theta}))\mathbf{1}_d$.

Remark 6. *Note that the framework (20) can be used for training explainers towards any probabilistic value. Intuitively, faster unbiased estimator substituted in the framework (20) would lead to better training of explainers. All in all, faster estimators could possibly benefit the research line of training semi-value-based explainers in feature attribution.*

Proposition 9. *Let* $\hat{\boldsymbol{\phi}}^{Sh} = (\mathbf{X}^\top\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{W}\left(\mathbf{U}_z - U_z(\emptyset)\mathbf{1}_{2^d-2}\right)$ *and* $\boldsymbol{\phi}^{Sh}$ *be the Shapley value using the utility function* U_z*. Then, there is*

$$\boldsymbol{\phi}^{Sh} = \hat{\boldsymbol{\phi}}^{Sh} + \rho\mathbf{1}_d$$

$$\textit{where } \rho = -\frac{\mathbf{1}_d^\top(\mathbf{X}^\top\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{W}(\mathbf{U}_z - U_z(\emptyset)\mathbf{1}_{2^d-2}) - U_z([d]) + U_z(\emptyset)}{\mathbf{1}_d^\top(\mathbf{X}^\top\mathbf{W}\mathbf{X})^{-1}\mathbf{1}_d}.$$

In other words, the target FastSHAP training explainers towards is potentially biased as ρ *is not necessarily zero.*

Proof. In view of the optimization problem (17), let $m_s \leftarrow \frac{d-1}{\binom{d}{s}s(d-s)}$, $n \leftarrow d$ and $U^n \leftarrow U_z$. By (Charnes et al. 1988, Theorem 4), the induced uniquely optimal solution $\mathbf{v}^*$ is exactly $\boldsymbol{\phi}^{Sh}$. According to Eq. (19), we have

$$\mathbf{v}^* = \mathbf{A}^{-1}\left(\mathbf{b} - \frac{\mathbf{1}_d^\top\mathbf{A}^{-1}\mathbf{b} - U_z([d]) + U_z(\emptyset)}{\mathbf{1}_d^\top\mathbf{A}^{-1}\mathbf{1}_d}\mathbf{1}_d\right)$$

$$\text{where } \mathbf{A} = \mathbf{X}^\top\mathbf{W}\mathbf{X} \text{ and } \mathbf{b} = \mathbf{X}^\top\mathbf{W}(\mathbf{U}_z - U_z(\emptyset)\mathbf{1}_{2^d-2}),$$

from which we can deduce

$$\boldsymbol{\phi}^{Sh} = \mathbf{v}^* = \mathbf{A}^{-1}\mathbf{b} + \rho\mathbf{1}_d = \hat{\boldsymbol{\phi}}^{Sh} + \rho\mathbf{1}_d$$

$$\text{where } \rho = -\frac{\mathbf{1}_d^\top\mathbf{A}^{-1}\mathbf{b} - U_z([d]) + U_z(\emptyset)}{\mathbf{1}_d^\top\mathbf{A}^{-1}\mathbf{1}_d}.$$

□

H INTERPRETATION OF $\kappa(N)$ AND $\tau(N)$ IN PROPOSITION 4

Recall that $\kappa(n) = \gamma(n)^{-2}$ where $\gamma(n) = \frac{\sum_{s=1}^{n-1}\binom{n-1}{s-1}(p_s^n + p_{s+1}^n)}{\sum_{s=1}^{n-1}\binom{n}{s}(p_s^n + p_{s+1}^n)}$. Our interpretation of $\kappa(n)$ is based on $\gamma(n)$. In Algorithm 1, for each non-empty proper subset S, its probability of being sampled is

$$P(S) = \frac{p_s^n + p_{s+1}^n}{\sum_{s=1}^{n-1}\binom{n}{s}(p_s^n + p_{s+1}^n)}.$$

Since $\gamma(n) = \sum_{S\subsetneq[n]\colon i\in S} P(S)$, $\gamma(n)$ is the probability that the i-th data point appears in a random sample S. Thanks to symmetry, the choice of i is immaterial. Thus,

$$n\gamma(n) = \sum_{i=1}^{n}\sum_{S\subsetneq[n]\colon i\in S} P(S) = \sum_{S\subsetneq[n]}\sum_{i=1}^{n}\mathbb{1}_{i\in S}P(S) = \sum_{\emptyset\subsetneq S\subsetneq[n]} sP(S) = \mathbb{E}_S[s],$$

whence follows

$$\gamma(n) = \frac{\mathbb{E}_S[s]}{n} \geq \frac{1}{n}.$$

The lower bound is achieved when $p_1^n = 1$ and $p_i^n = 0$ for all $i \geq 2$, corresponding to the semivalue with $\mu = \delta_0$, i.e., leave everything else out.

Looking into Algorithm 1, for each sampled utility evaluation $U^n(S)$, it is used to update the estimates of s data. Therefore, $\gamma(n)$ is just the average rate of reusing utility evaluations, and $\kappa(n)$ is the inverse square of this average rate. The higher $\gamma(n)$ is, the more efficient GELS-R is. On the flip side, reusing utility evaluations also creates correlation among the estimates. We conclude that for the benefit to outweigh the cost, $\gamma(n)$ needs to be on the order of $\omega(\frac{1}{\sqrt{n}})$, which is easily satisfied as long as the sequence $\{p_s^n\}$ is not exclusively concentrated around small s. All practical probabilistic values used in the literature, including the ones in our experiments, meet this condition.

Recall that $\tau(n) = \left(\sum_{s=1}^{n}\binom{n+1}{s}p_s^n\right)^2$ and we write $\zeta(n) = \tau(n)^{-\frac{1}{2}}$. Observe that, for every $i \in [n]$,

$$2\zeta(n) = \lambda \cdot \frac{1}{\sum_{s=1}^{n}\binom{n}{s}p_s^n} + (1-\lambda)\cdot\frac{1}{\sum_{s=1}^{n}\binom{n}{s-1}p_s^n}$$

where $\lambda = \frac{\sum_{s=1}^{n}\binom{n}{s}p_s^n}{\sum_{s=1}^{n}\binom{n+1}{s}p_s^n}$ is the probability of subsets sampled by Algorithm 2 not containing the introduced null data point (equivalently, the $(n+1)$-th data point). Then, for any $i \in [n]$,

$$\frac{1}{\sum_{s=1}^{n}\binom{n}{s}p_s^n} = \frac{\sum_{s=1}^{n}\binom{n-1}{s-1}p_s^n}{\sum_{s=1}^{n}\binom{n}{s}p_s^n} = \sum_{\emptyset\subsetneq S\subseteq[n]\colon i\in S} P_{\not\ni n+1}(S)$$

where $P_{\not\ni n+1}(S)$ is the probability of the set S sampled by Algorithm 2 conditioned on the samples do not contain the $(n+1)$-th data point. Therefore,

$$\frac{n}{\sum_{s=1}^{n}\binom{n}{s}p_s^n} = \sum_{i=1}^{n}\sum_{\emptyset\subsetneq S\subseteq[n]}\mathbb{1}_{i\in S}P_{\not\ni n+1}(S) = \sum_{\emptyset\subsetneq S\subseteq[n]} sP_{\not\ni n+1}(S) = \mathbb{E}_{S|S\not\ni n+1}[s].$$

On the other hand, for every $i \in [n]$,

$$\frac{1}{\sum_{s=1}^{n}\binom{n}{s-1}p_s^n} = \frac{\sum_{s=1}^{n}\binom{n-1}{s-1}p_s^n}{\sum_{s=1}^{n}\binom{n}{s-1}p_s^n} = \sum_{\emptyset\subsetneq S\subsetneq[n+1]\colon i\notin S, n+1\in S} P_{\ni n+1}(S),$$

which leads to

$$\begin{aligned}\frac{n}{\sum_{s=1}^{n}\binom{n}{s-1}p_s^n} &= \sum_{i=1}^{n}\sum_{\emptyset\subsetneq S\subsetneq[n+1]\colon n+1\in S}\mathbb{1}_{i\notin S}P_{\ni n+1}(S)\\ &= \sum_{\emptyset\subsetneq S\subsetneq[n+1]\colon n+1\in S}(n+1-s)P_{\ni n+1}(S) = \mathbb{E}_{S|S\ni n+1}[n+1-s].\end{aligned}$$

Eventually, we have

$$\begin{aligned}\zeta(n) &= \frac{1}{2n}\left(\lambda \cdot \mathbb{E}_{S \mid S \not\ni n+1}[s] + (1-\lambda)\cdot \mathbb{E}_{S \mid S \ni n+1}[n+1-s]\right) \\ &= \frac{1}{2n}\mathbb{E}_S[s\mathbb{1}_{n+1 \notin S} + (n+1-s)\mathbb{1}_{n+1 \in S}].\end{aligned}$$

Looking into Algorithm 2, i) if the sampled subset S does not contain the null data, there are s of $\{\hat{r}_i - \hat{r}_{n+1}\}_{1 \leq i \leq n}$ receiving updates from $U^n(S)$; ii) for the other way, there are $n+1-s$ of them receiving updates from $U^n(S)$. To conclude, $\frac{\tau(n)}{4}$ is the square inverse of the average rate of reusing utility evaluations while running Algorithm 2.

I ASYMPTOTIC ANALYSIS

This appendix is mainly to analyze the asymptotic behavior (as $n \to \infty$) of $\kappa(n)$ and $\tau(n)$ that appear in Proposition 4. For probabilistic values, generally, there is no restriction on how $\{\mathbf{p}^n \in \mathbb{R}^n\}_{n \geq 1}$ are organized. Therefore, to analyze the convergence of our proposed estimators, we focus on semi-values instead. According to Dubey et al. (1981), each semi-value corresponds to a probability measure μ on the interval $[0, 1]$ such that

$$p_s^n = \int_0^1 t^{s-1}(1-t)^{n-s}\mathrm{d}\mu(t) \text{ for every } 1 \leq s \leq n. \tag{21}$$

Lemma 2. *Suppose $n > 1$, and for each probability measure μ on the closed interval $[0, 1]$, define*

$$m_k^\mu = \int_0^1 t^k \mathrm{d}\mu(t) \text{ and } w_k^\mu = \int_0^1 (1-t)^k \mathrm{d}\mu(t) \text{ for every } k \geq 0,$$

$$M_k^\mu = \sum_{j=0}^k m_k^\mu \text{ and } W_k^\mu = \sum_{j=0}^k w_k^\mu \text{ for every } k \geq 0.$$

Then, there is

$$\kappa(n)^{\frac{1}{2}} = \frac{M_{n-2}^\mu + W_{n-2}^\mu}{M_{n-2}^\mu} \text{ and } \tau(n)^{\frac{1}{2}} = M_{n-1}^\mu + W_{n-1}^\mu.$$

Proof. Recall that in Proposition 4 $\kappa(n)^{\frac{1}{2}} = \frac{\sum_{s=1}^{n-1}\binom{n}{s}(p_s^n + p_{s+1}^n)}{\sum_{s=1}^{n-1}\binom{n-1}{s-1}(p_s^n + p_{s+1}^n)}$. By Eq. (21), there is $p_s^n + p_{s+1}^n = p_s^{n-1}$ for every $1 \leq s \leq n-1$. Notice that

$$\sum_{s=1}^{n-1}\binom{n}{s}p_s^{n-1} = \sum_{s=1}^{n-1}\binom{n-1}{s}p_s^{n-1} + \sum_{s=1}^{n-1}\binom{n-1}{s-1}p_s^{n-1}.$$

By Eq. (21),

$$\begin{aligned}\sum_{s=1}^{n-1}\binom{n-1}{s}p_s^{n-1} &= \int_0^1 \sum_{s=1}^{n-1}\binom{n-1}{s}t^{s-1}(1-t)^{n-1-s}\mathrm{d}\mu(t) \\ &= \int_0^1 \frac{1}{t}\sum_{s=1}^{n-1}\binom{n-1}{s}t^s(1-t)^{n-1-s}\mathrm{d}\mu(t) \\ &= \int_0^1 \frac{1-(1-t)^{n-1}}{t}\mathrm{d}\mu(t) = \int_0^1 \frac{(1-(1-t))(\sum_{j=0}^{n-2}(1-t)^j)}{t}\mathrm{d}\mu(t) \\ &= W_{n-2}^\mu\end{aligned}$$

Note $\sum_{s=1}^{n-1}\binom{n-1}{s}t^{s-1}(1-t)^{n-1-s} = \sum_{j=0}^{n-2}(1-t)^j$ still holds for $t = 0$. Similarly, one can get

$$\sum_{s=1}^{n-1}\binom{n-1}{s-1}p_s^{n-1} = M_{n-2}^\mu.$$

Recall that in Proposition 4 $\tau(n)^{\frac{1}{2}} = \sum_{s=1}^n \binom{n+1}{s}p_s^n$, and thus one can get $\tau(n)^{\frac{1}{2}} = M_{n-1}^\mu + W_{n-1}^\mu$ in a similar fashion. □

Using the monotone convergence theorem, we have

$$\begin{aligned}\lim_{n\to\infty} M_n^\mu &= \lim_{n\to\infty}\int_0^1 \frac{1-t^{n+1}}{1-t}\mathrm{d}\mu(t) = \int_0^1 \lim_{n\to\infty}\frac{1-t^{n+1}}{1-t}\mathrm{d}\mu(t) = \int_0^1 \frac{1}{1-t}\mathrm{d}\mu(t),\\ \lim_{n\to\infty} W_n^\mu &= \lim_{n\to\infty}\int_0^1 \frac{1-(1-t)^{n+1}}{t}\mathrm{d}\mu(t) = \int_0^1 \frac{1}{t}\mathrm{d}\mu(t).\end{aligned} \tag{22}$$

Note that $\lim_{n\to\infty} M_n^{\delta_1} = \int_0^1 \frac{1}{1-t}\mathrm{d}\delta_1(t)$ is trivially true. Therefore, $\tau(n)\in\Theta(1)$ if and only if the two integrals in the above are finite. In particular, It holds if the probability density function p of μ satisfies $\lim_{t\to 0}\frac{p(t)}{t^a} = \lim_{t\to 1}\frac{p(t)}{(1-t)^b} = 0$ for some $a, b > 0$; see Proposition 10. Interestingly, we have

$$\kappa(n)^{\frac{1}{2}} = 1 + \frac{W_{n-2}^\mu}{M_{n-2}^\mu} \le 1 + W_{n-2}^\mu.$$

Thus, $\kappa(n) = \Theta(1)$ if the integral Eq. (22) is finite, meaning that μ cannot put "large" mass around $t = 0$. However, this is not necessary as any symmetric probability measure μ, e.g., the uniform distribution that leads to the Shapley value, will have $\kappa(n) = 4$.

Remark 7. *For extreme cases $\mu = \delta_0$ (leave everything else out) and $\mu = \delta_1$ (leave one out), it is clear that $\kappa(n)\in\Theta(1)$ for the latter but it becomes $\Theta(n^2)$ for the former, in which case our GELS-R (Algorithm 1) in fact only samples subsets of size 1, and therefore each utility evaluation is used to update the estimate of one data point. By contrast, the (weighted) sampling lift estimator always spends two utility evaluations to update the estimate of one data point.*

Corollary 1. *For the Banzhaf value, $\kappa(n), \tau(n)\in\Theta(1)$. In other words, GELS, GELS-R and GELS-Shapley require $O(\frac{n}{\epsilon^2}\log\frac{n}{\delta})$ utility evaluations to achieve an (ϵ,δ)-approximation for the Banzhaf value.*

Proof. For the Banzhaf value, $\mu = \delta_{0.5}$ (Dirac delta distribution), and thus $\lim_{n\to\infty} M_n^{\delta_{0.5}} = \lim_{n\to\infty} W_n^{\delta_{0.5}} = 2$. ☐

We now provide some easily verifiable conditions that determine the growth of $\kappa(n)$ and $\tau(n)$. Recall that as long as $\kappa(n), \tau(n) = o(n)$, our estimators are more efficient than the sampling lift estimator.

Proposition 10. *Assume the probability measure μ admits a density function p such that $\mu(S) = \int_S p(t)\mathrm{d}t$ for every Borel-measurable subset $S\subseteq[0,1]$, then,*

1. *$\kappa(n), \tau(n)\in\Theta(1)$ if there exist $a, b > 0$ such that $\lim_{t\to 0}\frac{p(t)}{t^a} = \lim_{t\to 1}\frac{p(t)}{(1-t)^b} = 0$.*
2. *$\kappa(n), \tau(n)\in O(\log(n)^2)$ if $\limsup_{t\to 0} p(t) < \infty$ and $\limsup_{t\to 1} p(t) < \infty$. Examples include if p is bounded (the so-called continuous semivalues used by Dubey et al. (1981)).*

Proof. We first show that there exist counterexamples if these conditions are violated. For the Shapley value with $\mu = \mathcal{U}$ (the uniform measure), $m_k^{\mathcal{U}} = w_k^{\mathcal{U}} = \frac{1}{k+1}$, and thus $\tau(n) = \Theta(\log(n)^2)$. For the second, looking into Lemma 3, consider $\alpha = \beta < 1$, $\kappa(n), \tau(n)\in\Theta(n^{1-\alpha})$.

Suppose there exist $a, b > 0$ such that $\lim_{t\to 0}\frac{p(t)}{t^a} = \lim_{t\to 1}\frac{p(t)}{(1-t)^b} = 0$. Then, there exists some $\epsilon, C > 0$ such that $p(t)\le Ct^a(1-t)^b$ if $t < \epsilon$ and $t > 1-\epsilon$. Define a positive measure μ_ϵ by letting $\mu_\epsilon(S) = \mu(S\cap[\epsilon, 1-\epsilon])$ for every Borel-measurable subset $S\subseteq[0,1]$. Besides, define another positive measure λ by letting $\lambda(S) = \int_S Ct^a(1-t)^b\mathrm{d}t$ for every Borel-measurable subset $S\subseteq[0,1]$. Therefore, we have $\mu\le\mu_\epsilon + \lambda$, which indicates that $m_k^\mu \le m_k^{\mu_\epsilon} + m_k^\lambda$, and thus $M_k^\mu \le M_k^{\mu_\epsilon} + M_k^\lambda$. By Lemma 3, $M_k^\lambda\in\Theta(1)$. For the other, observe that $m_k^{\mu_\epsilon}\le m_k^{\delta_{1-\epsilon}}$, and therefore $M_k^{\mu_\epsilon}\le M_k^{\delta_{1-\epsilon}}\in\Theta(1)$. The remaining case W_k^μ can be tackled similarly. To conclude, $\kappa(n), \tau(n)\in\Theta(1)$.

Suppose $\limsup_{t\to 0} p(t) < \infty$ and $\limsup_{t\to 1} p(t) < \infty$, then there exist $\epsilon, C > 0$ such that $p(t)\le C$ for every $t\in[0,\epsilon)\cup(1-\epsilon, 1]$. Define μ_ϵ as the one in the above. Then, $\mu\le\mu_\epsilon + C\mathcal{U}$

where $\mathcal{U}$ denotes the uniform measure. Therefore, $w_k^{\mu} \leq w_k^{\mu_\epsilon} + w_k^{C\mathcal{U}}$. Since $w_k^{C\mathcal{U}} = \frac{C}{k+1}$, there is $W_k^{C\mathcal{U}} \in \Theta(\log n)$. Besides, $W_k^{\mu_\epsilon} \leq W_k^{\delta_\epsilon} \in \Theta(1)$. The remaining case M_k^{μ} can be dealt with similarly. Therefore, the conclusion follows by using Lemma 2. $\square$

Next, we derive a more precise estimate for the Beta Shapley values.

Lemma 3. *Let $B(\alpha, \beta)$ be the beta distribution with probability density function $\propto t^{\alpha-1}(1-t)^{\beta-1}$. Note that if $\mu = B(\alpha, \beta)$, it yields the Beta(β, α) (a parameterized Beta Shapley value).*

$$M_n^{B(\alpha,\beta)} \in \begin{cases} \Theta(\log n) & \beta = 1 \\ \Theta(1) & \beta > 1 \\ \Theta(n^{1-\beta}) & 0 < \beta < 1 \end{cases} \quad \textit{and } W_n^{B(\alpha,\beta)} \in \begin{cases} \Theta(\log n) & \alpha = 1 \\ \Theta(1) & \alpha > 1 \\ \Theta(n^{1-\alpha}) & 0 < \alpha < 1 \end{cases}.$$

Proof. Let Γ denote the Gamma function.

$$m_k^{\mathrm{B}(\alpha,\beta)} = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \cdot \frac{\Gamma(\alpha+k)\Gamma(\beta)}{\Gamma(\alpha+\beta+k)} = \frac{\prod_{j=0}^{k-1}(\alpha+j)}{\prod_{j=0}^{k-1}(\alpha+\beta+j)}.$$

We write $a_k \sim b_k$ if $\lim_{k\to\infty} \frac{a_k}{b_k}$ converges. Since $\Gamma(x) = \lim_{n\to\infty} \frac{n! n^x}{x(x+1)\cdots(x+n)}$, e.g., see (Rudin 1953, Eq. (95) in Chapter 8), we have $m_k^{\mathrm{B}(\alpha,\beta)} \sim \frac{k! k^{\alpha}}{k! k^{\alpha+\beta}} = k^{-\beta}$, and thus the conclusion follows. The remaining case can be derived similarly. $\square$

Remark 8. *Lemma 3 is useful for obtaining the time complexity of the proposed estimators for the Beta Shapley values parameterized by $\alpha, \beta > 0$. If $\alpha, \beta > 1$, the two proposed estimators require $O(\frac{n}{\epsilon^2}\log\frac{n}{\delta})$ utility evaluations to achieve an (ϵ, δ)-approximation. Note that this is the currently best time complexity. For the Shapley value, i.e., Beta$(1,1)$, it is $O(\frac{n}{\epsilon^2}\log\frac{n}{\delta})$ for GELS-R, and $O(\frac{n}{\epsilon^2}\log(\frac{n}{\delta})\log(n)^2)$ for GELS and GELS-Shapley. To our knowledge, in terms of (ϵ, δ)-approximation, the previously best time complexity for the Shapley value is $O(\frac{n}{\epsilon^2}\log(\frac{n}{\delta})\log n)$ achieved by the group testing estimator (Wang and Jia 2023, Theorem C.7). Therefore, GELS-R provides the currently best time complexity in terms of (ϵ, δ)-approximation for the Shapley value.*

J MORE EXPERIMENT RESULTS

For training estimators on MNIST, all performance curves are provided in Figure 3. It can be seen that the conclusions we have in the main paper still hold on MNIST.

The paired sampling (PS) technique was proposed by Covert and Lee (2021) to enhance (unbiased) KernelSHAP, but we notice that it can also be employed for GELS, GELS-R, GELS-Shapley, (weighted) sampling lift, group testing, AME, MSR. Precisely, suppose the sampled subsets is $\{S_k\}_{k\geq 1}$, the PS technique employs sampled subsets $\{T_j\}_{j\geq 1}$ such that $T_{2k-1} = S_k$ and $T_{2k} = [n] \backslash S_k$ for every $k \geq 1$. Roughly speaking, symmetric probabilistic values, i.e., $p_s^n = p_{n-s}^n$ for every $s \in [n]$ could possibly takes advantage of this technique. Examples include the Shapley value, the Banzhaf value and Beta(γ, γ). An exception is that weighted sampling lift can be coupled with PS for any probabilistic value. To have fair comparison, we implement the PS technique if possible. The results with utility functions reporting the classification accuracy on D_{perf} are shown in Figures 4, 5, 6, 7 and 8. Moreover, we also set the utility functions to report the cross entropy loss instead, and the corresponding results are presented in Figures 9, 10, 11, 12 and 13.

Surprisingly, AME and group testing gain significant performance boosts using the PS technique, while (unbiased) KernelSHAP takes advantage of this technique occasionally. For other estimators, PS does not play a noticeable role. Interestingly, group testing with PS is exactly equal to GELS with PS, whereas GELS-Shapley with PS is exactly equal to unbiased KernelSHAP with PS.

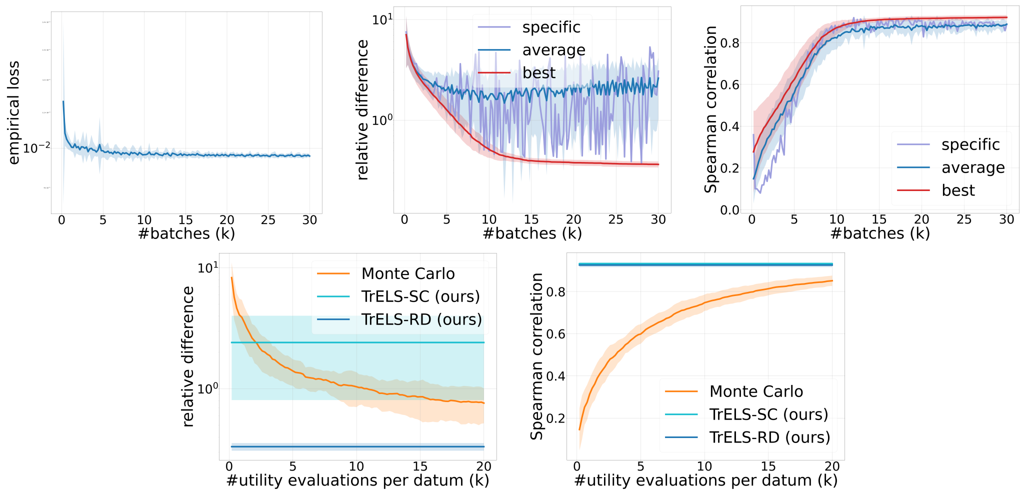
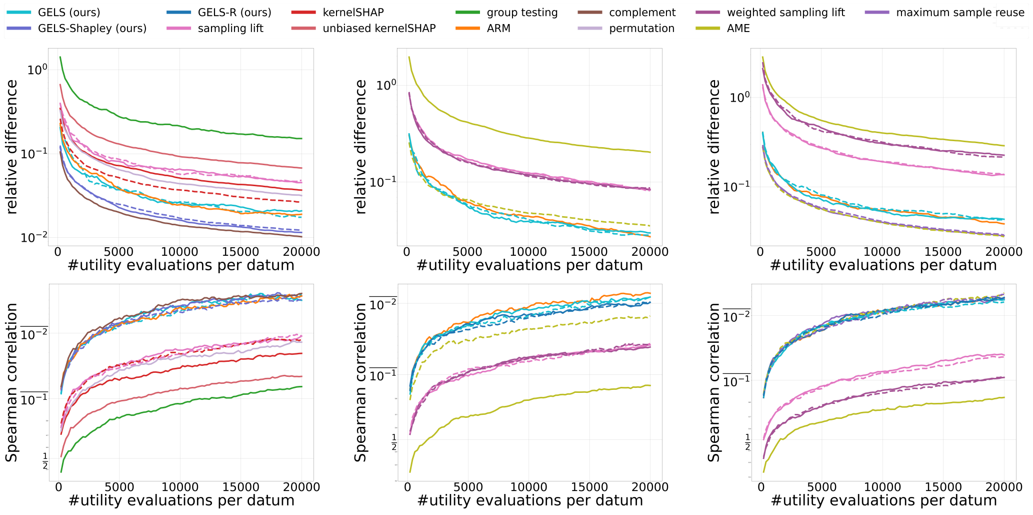
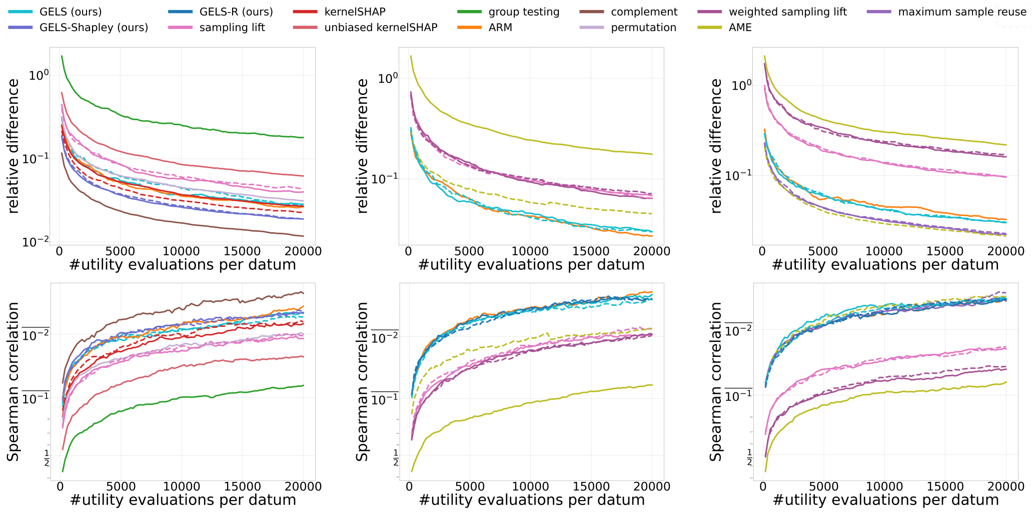


Figure 3: i) The first row: The first plot is the empirical loss of the sampled batch, while the others present the relative distance $\|\hat{\varphi} - \varphi\|_2 / \|\varphi\|_2$ and the Spearman correlation between φ and $\hat{\varphi}$. $\hat{\varphi}$ denotes the one predicted by the estimators trained on MNIST. The first two plots are in log scale. “best” reports the best one achieved by the trained estimators at each time point, whereas “specific” is the one with the random seed being 0. ii) The second row: the two plots compare the performance of the estimators trained on MNIST with Monte-Carlo method using Eq (2). Specifically, TrELS-SC uses trained estimators that achieve the best Spearman correlation on D_{val}, whereas RD stands for relative difference.



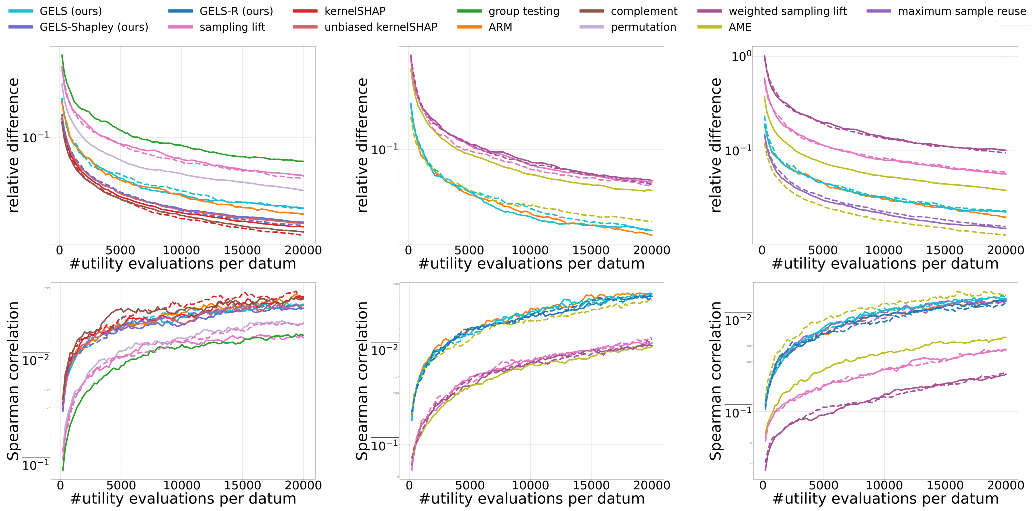
Shapley
Beta(2, 2)
Banzhaf

Figure 4: Comparison of different estimators on probabilistic values using wind. The relative difference is plotted in log scale while the Spearman correlation is in logit scale. In addition, $\overline{r} = 1 - r$. The utility functions reports the classification accuracy on D_{val}.



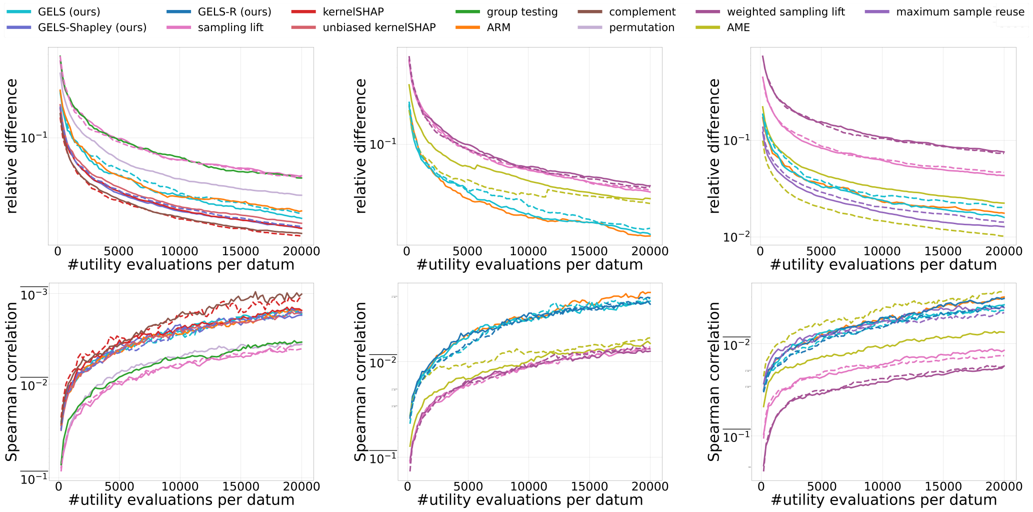
Shapley Beta(2, 2) Banzhaf

Figure 5: Comparison of different estimators on probabilistic values using iris. The relative difference is plotted in log scale while the Spearman correlation is in logit scale. In addition, $\overline{r} = 1 - r$. The utility functions reports the classification accuracy on D_{val}.



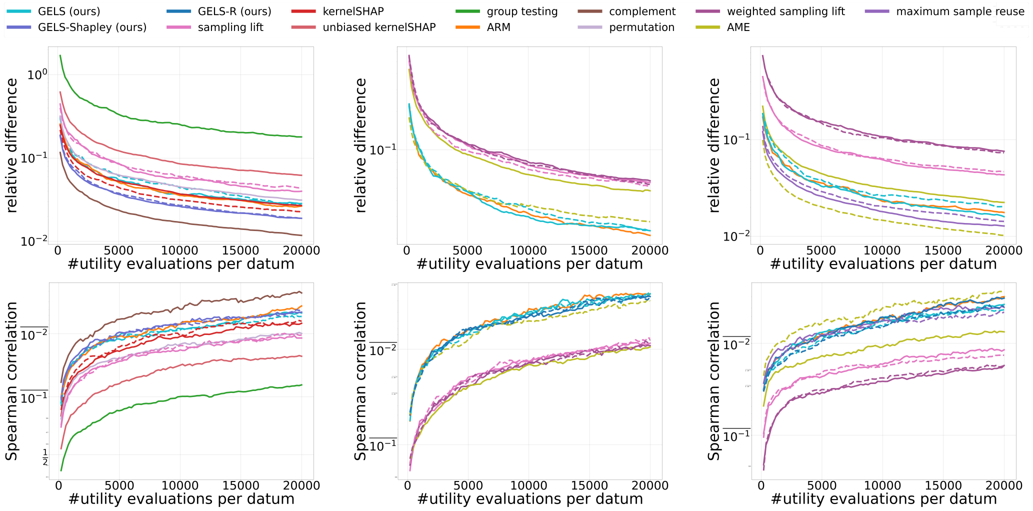
Shapley Beta(2, 2) Banzhaf

Figure 6: Comparison of different estimators on probabilistic values using MNIST. The relative difference is plotted in log scale while the Spearman correlation is in logit scale. In addition, $\overline{r} = 1 - r$. The utility functions reports the classification accuracy on D_{val}.



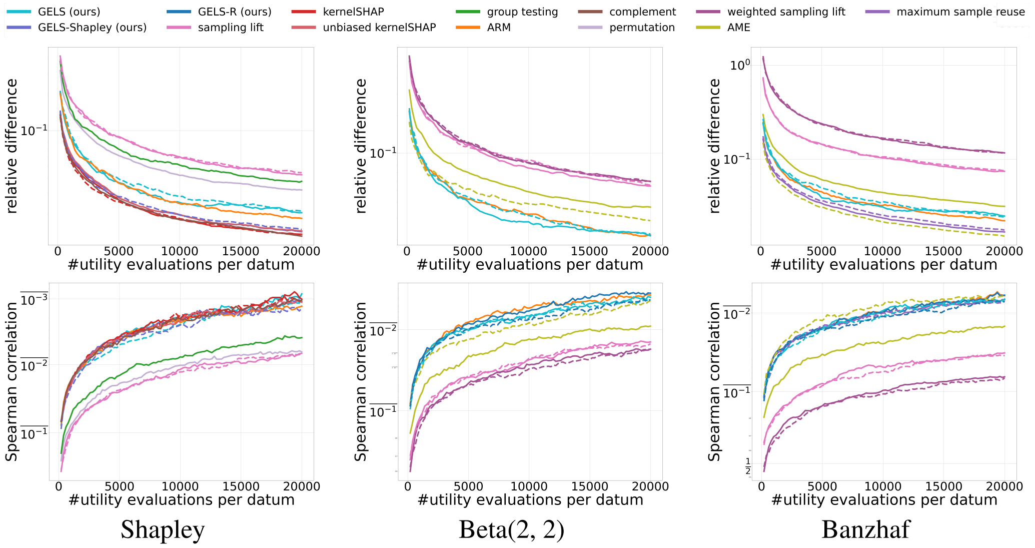
Shapley Beta(2, 2) Banzhaf

Figure 7: Comparison of different estimators on probabilistic values using FMNIST. The relative difference is plotted in log scale while the Spearman correlation is in logit scale. In addition, $\overline{r} = 1 - r$. The utility functions reports the classification accuracy on D_{val}.



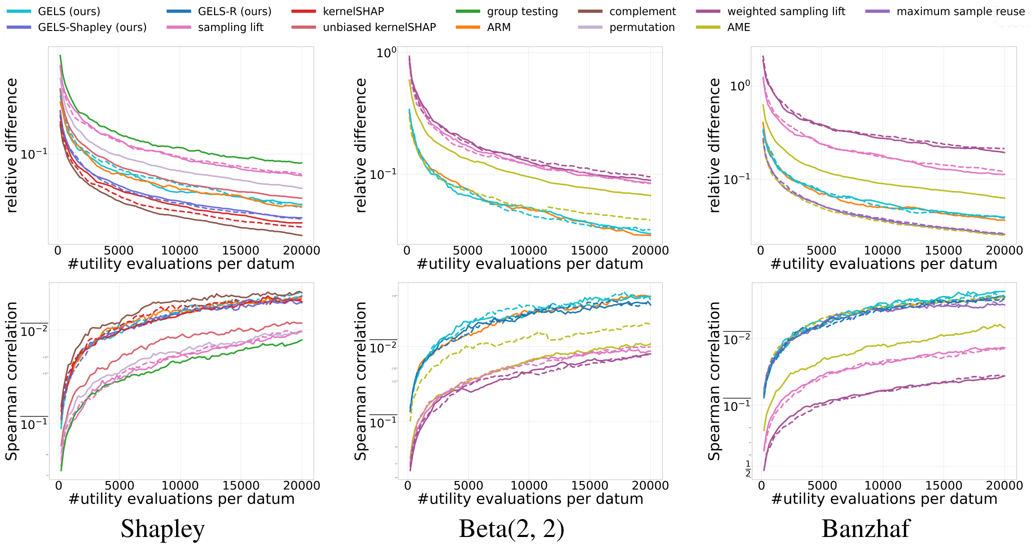
iris MNIST FMNIST

Figure 8: Comparison of different estimators on different datasets while using Beta$(4, 1)$. The relative difference is plotted in log scale while the Spearman correlation is in logit scale. In addition, $\overline{r} = 1 - r$. The utility functions reports the classification accuracy on D_{val}.



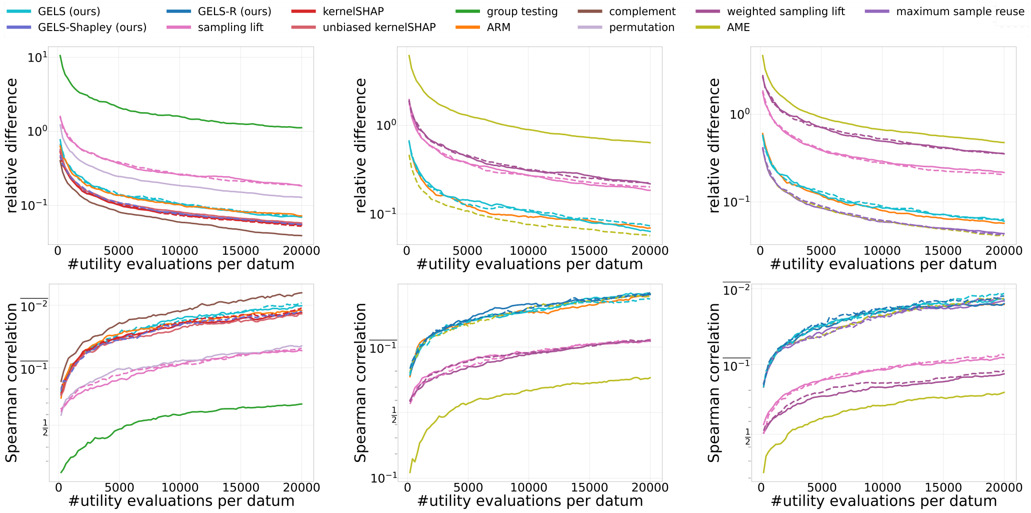
Shapley Beta(2, 2) Banzhaf

Figure 9: Comparison of different estimators on probabilistic values using wind. The relative difference is plotted in log scale while the Spearman correlation is in logit scale. In addition, $\overline{r} = 1 - r$. The utility functions reports the cross entropy loss on D_{val}.



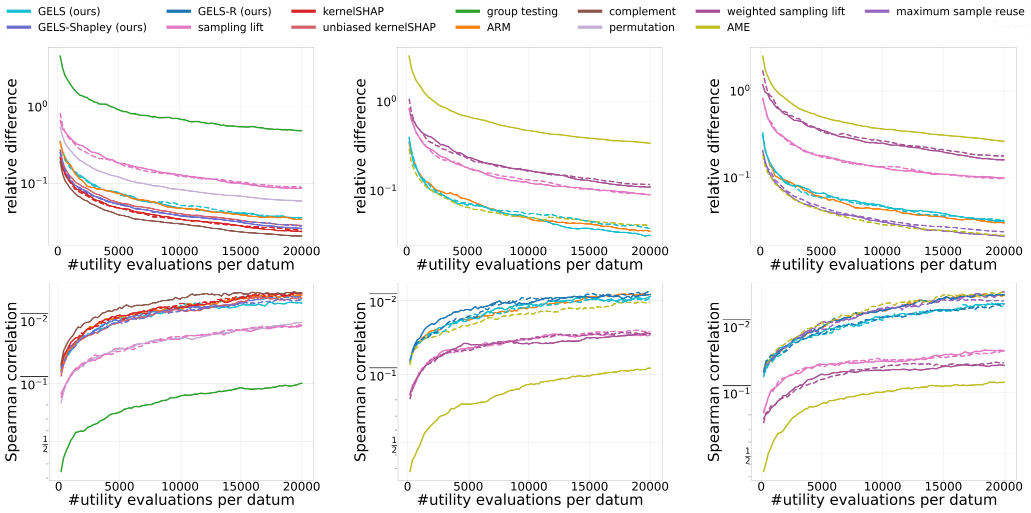
Shapley Beta(2, 2) Banzhaf

Figure 10: Comparison of different estimators on probabilistic values using iris. The relative difference is plotted in log scale while the Spearman correlation is in logit scale. In addition, $\overline{r} = 1 - r$. The utility functions reports the cross entropy loss on D_{val}.



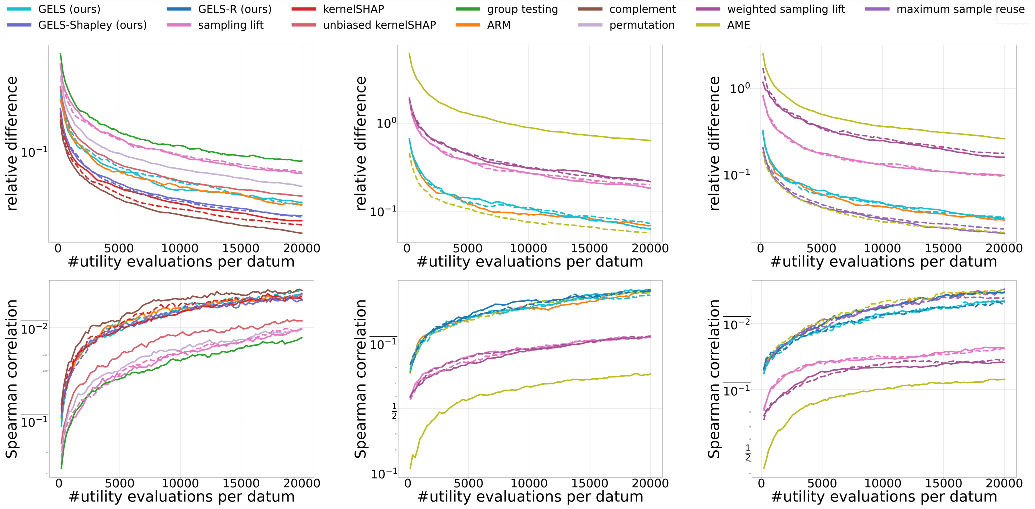
Shapley Beta(2, 2) Banzhaf

Figure 11: Comparison of different estimators on probabilistic values using MNIST. The relative difference is plotted in log scale while the Spearman correlation is in logit scale. In addition, $\overline{r} = 1 - r$. The utility functions reports the cross entropy loss on D_{val}.



Shapley Beta(2, 2) Banzhaf

Figure 12: Comparison of different estimators on probabilistic values using FMNIST. The relative difference is plotted in log scale while the Spearman correlation is in logit scale. In addition, $\overline{r} = 1 - r$. The utility functions reports the cross entropy loss on D_{val}.



iris MNIST FMNIST

Figure 13: Comparison of different estimators on different datasets while using $\mathrm{Beta}(4, 1)$. The relative difference is plotted in log scale while the Spearman correlation is in logit scale. In addition, $\overline{r} = 1 - r$. The utility functions reports the cross entropy loss on D_{val}.