Explaining Preferences with Shapley Values

Robert Hu[*†] Amazon London

Siu Lun Chau[*] Department of Statistics University of Oxford

Jaime Ferrando Huertas Shaped New York

Dino Sejdinovic Department of Statistics University of Oxford

Abstract

While preference modelling is becoming one of the pillars of machine learning, the problem of preference explanation remains challenging and underexplored. In this paper, we propose PREF-SHAP, a Shapley value-based model explanation framework for pairwise comparison data. We derive the appropriate value functions for preference models and further extend the framework to model and explain *context specific* information, such as the surface type in a tennis game. To demonstrate the utility of PREF-SHAP, we apply our method to a variety of synthetic and real-world datasets and show that richer and more insightful explanations can be obtained over the baseline.

1 Introduction

Preference learning [1] is a classical problem in machine learning, where one is interested in learning the order relations on a collection of data items. Preference learning algorithms [2–5] often assume that there is a latent utility function $f : \mathcal{X} \mapsto \mathbb{R}$ dictating the outcome of preferences, where $\mathcal{X}$ denotes the domain of item covariates. An explicit feedback such as item ratings or rankings from recommender systems can be treated as noisy evaluations of f, whereas pairwise comparison data (also known as duelling data) arising from, e.g., sports match outcomes [6, 7] can be used to implicitly infer f, i.e. item $\mathbf{x}^{(\ell)}$ is preferred over (beats) item $\mathbf{x}^{(r)}$ when $f(\mathbf{x}^{(\ell)}) > f(\mathbf{x}^{(r)})$. As shown by Kahneman and Tversky [8], humans often struggle with evaluating absolute quantities when it comes to eliciting preferences, but are broadly capable of evaluating relative differences, a core observation often exploited in preference learning. Motivated by such, this work will focus on explaining preferences inferred using duelling data.

Explaining preference models is crucial when they are applied in areas such as recommendation systems [9], finance [10], and sports science [11] for the practitioner to trust, debug and understand the value of their findings [12]. However, despite its importance, no prior work has studied this problem to the best of our knowledge. While one may suggest applying existing explainability tools such as LIME [13], or SHAP [14] to a learned utility function f, we reason that this approach only explains the utility but not the mechanism of eliciting preferences itself. We highlight the important differences between these two viewpoints in our numerical experiments. Moreover, the utility-based model places a strong *rankability* assumption on the underlying preferences, meaning that if we define $\mathbf{x}^{(\ell)} \preceq \mathbf{x}^{(r)} \iff f(\mathbf{x}^{(\ell)}) \leq f(\mathbf{x}^{(r)})$, then $\preceq$ is a total order on all the items. However, as Pahikkala et al. [15] and Chau et al. [16] have discussed, there are many departures from rankability in practice, e.g. we might easily see a preference of A over B, B over C, but C over A – conforming to the *rock-paper-scissors* relation. Such inconsistent preferences are under frequent study in social choice theory [17, 18], and are of wider interest in both healthcare [19] and retail [20] where data are both large and noisy.

To move beyond the rankability assumption, we will utilise the *Generalised Preferential Kernel* from [16] to model the underlying preferences, and develop PREF-SHAP, a novel Shapley value [21]-based

[*]Equal contribution, order decided by coinflip

[†]Work primarily done at the University of Oxford and finished at Amazon.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

explainability toolbox, to explain the inferred preferences. Our contributions can be summarised as follows:

1. We propose PREF-SHAP, a novel Shapley value-based explainability algorithm, to explain preferences based on duelling data.
2. We empirically demonstrate that PREF-SHAP gives more informative explanations compared to the naive approach of applying SHAP to the inferred utility function f.
3. We release a high-performant implementation of PREF-SHAP at [22].

2 Background materials

We will first give a brief overview of preference learning and Shapley Additive Explanations (SHAP) [14], which are the two core concepts of our contribution, PREF-SHAP, described in Section 3.

Notation Scalars are denoted by lower case letters, while vectors and matrices are denoted by bold lower case and upper case letters, respectively. Random variables are denoted by upper case letters. $\mathcal{X} \subseteq \mathbb{R}^d$ denotes the item space with d features and $\mathcal{Y} = \{-1, 1\}$ is the binary preference outcome space[3]. We let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel function and $\mathcal{H}_k$ the corresponding reproducing kernel Hilbert space (RKHS).

2.1 Preference Learning

In this section, we will introduce the two approaches to model preferences from duelling data, namely the *utility based approach* and the more general approach from Chau et al. [16]. Formally, a preference feedback is denoted as *duelling*, when a pair of items $(\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}) \in \mathcal{X} \times \mathcal{X}$ is given to a user, and a binary outcome $y \in \mathcal{Y}$ telling us whether $\mathbf{x}^{(\ell)}$ or $\mathbf{x}^{(r)}$ won the duel, is observed. In general, we observe m binary preferences among n items, giving the data $D = \left(\mathbf{y}, \mathbf{X}^{(\ell)}, \mathbf{X}^{(r)}\right) = \left\{(y_j, \mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)})\right\}_{j=1}^{m}$. We also use $\mathbf{X} \in \mathbb{R}^{n \times d}$ to denote the full item covariate matrix.

Utility-based Preference model (UPM) The following likelihood model is often used [2–5, 7] to model duelling feedback using a latent utility function f:

$$p\left(y \mid \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right) = \sigma\left(y\left(f\left(\mathbf{x}^{(\ell)}\right) - f\left(\mathbf{x}^{(r)}\right)\right)\right), \tag{1}$$

where σ is the logistic CDF, i.e. $\sigma(z) = (1 + \exp(-z))^{-1}$. Maximum likelihood approaches are then deployed to learn the latent utility function f. Consequently, preferences between items can be inferred accordingly from $\mathbf{f} = \{f(\mathbf{x}_i)\}_{i=1}^{n}$, i.e. $\mathbf{x}_i$ is on average preferred over $\mathbf{x}_j$ if $\mathbf{f}_i \geq \mathbf{f}_j$.

Albeit elegant, there are several drawbacks to this approach in modelling preferences. As mentioned, using a one-dimensional vector $\mathbf{f}$ to derive preferences assumes that the items $\{\mathbf{x}_i\}_{i=1}^{n}$ are perfectly rankable, i.e. there is a total ordering on $\mathcal{X}$ which the true preferences are consistent with. This is a strong assumption that often does not hold in practice. For example, it is well studied that cognitive biases often lead to inconsistent human preferences in behavioural economics [8]. Moreover, the ranking community has also challenged this assumption by devising rankability metrics [23, 24] to test this restrictive assumption in practice.

Generalised Preference Model (GPM) Chau et al. [16] proposed to model preference directly using a more general $g : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that captures the preference within any pair of items, using the likelihood

$$p\left(y \mid \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right) = \sigma\left(yg\left(\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right)\right). \tag{2}$$

We note that g has to be a skew-symmetric function to ensure the natural property $p(y \mid \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}) = 1 - p(y \mid \mathbf{x}^{(r)}, \mathbf{x}^{(\ell)})$. The utility based approach can be obtained as a special case of this model, i.e. by setting $g(\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}) = f(\mathbf{x}^{(\ell)}) - f(\mathbf{x}^{(r)})$. We propose that when one is interested in modelling (and thus explaining) pairwise preferences, we should consider the preference function g directly instead of explaining preferences based on a restrictive utility model f.

[3] Thus, we do not model 'draws' in match outcomes, but the model can be straightforwardly extended to include them by specifying the appropriate likelihood function.

We follow Chau et al. [16]'s approach to model g non-parametrically using kernel methods [25]. We assume g as a function lives in the following RKHS of skew-symmetric functions: given kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined on the item space $\mathcal{X}$, the *generalised preferential kernel* k_E on $\mathcal{X} \times \mathcal{X}$ is constructed as follows:

$$k_E\left(\left(\mathbf{x}_i^{(\ell)}, \mathbf{x}_i^{(r)}\right), \left(\mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}\right)\right) = k\left(\mathbf{x}_i^{(\ell)}, \mathbf{x}_j^{(\ell)}\right) k\left(\mathbf{x}_i^{(r)}, \mathbf{x}_j^{(r)}\right) - k\left(\mathbf{x}_i^{(\ell)}, \mathbf{x}_j^{(r)}\right) k\left(\mathbf{x}_i^{(r)}, \mathbf{x}_j^{(\ell)}\right).$$

This kernel allows us to model the similarity across pairs of items. Moreover, if k is a universal kernel [26], then k_E also satisfies the corresponding notion of universality, meaning that the corresponding RKHS $\mathcal{H}_{k_E}$ is rich enough to approximate any bounded continuous skew-symmetric function arbitrarily well [16, Theorem. 1]. To infer $g \in \mathcal{H}_{k_E}$ using likelihood (2), one simply runs kernel logistic regression with data $\mathbf{y}$ as labels and $\left(\mathbf{X}^{(\ell)}, \mathbf{X}^{(r)}\right)$ as inputs. We will refer to this approach as the *Generalised Preference Model* (GPM).

We emphasize that *explaining* GPM allows us to specifically explain *inconsistent preferences*, which in contrast to *explaining rank* allows us to infer preferences even when transitivity is violated. Such insights can be of great importance in broader contexts such as decision theory [27] and utility theory [28] where transitivity does not hold.

Incorporating context variables. Besides item-level covariates $\mathbf{x} \in \mathcal{X}$, when there exist additional *context covariates* $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^{d'}$ that describe the context in which a specific pairwise comparison is made, they can be incorporated into the kernel design as discussed in Chau et al. [16, Appendix. B]. Examples of such context covariates could be court type when a tennis match is conducted, or where a different user compares two clothing items in e-commerce. Considering the enriched dataset $D = \left\{\left(y_j, \mathbf{u}_j, \mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}\right)\right\}_{j=1}^m$, we can now model the preference incorporating the context as: $p(y \mid \mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}) = \sigma\left(g_U\left(\mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right)\right)$. Now, given a kernel k_U defined on the context space $\mathcal{U}$, the context-specific preference function $g_U : \mathcal{U} \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ can be learnt non-parametrically with the following kernel,

$$k_E^{(U)}\left(\left(\mathbf{u}_i, \mathbf{x}_i^{(\ell)}, \mathbf{x}_i^{(r)}\right), \left(\mathbf{u}_j, \mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}\right)\right) = k_U\left(\mathbf{u}_i, \mathbf{u}_j\right) k_E\left(\left(\mathbf{x}_i^{(\ell)}, \mathbf{x}_i^{(r)}\right), \left(\mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}\right)\right).$$

We refer to this approach as the *Context-specific Generalised Preference Model* (C-GPM).

2.2 Shapley Additive Explanations (SHAP)

To explain preferences, we will utilise the popular SHAP (SHapley Additive exPlanations) paradigm, which is based on the concept of Shapley values (SV). SV [21] were originally proposed as a credit allocation scheme for a group of d players in the context of cooperative games, which are characterised by a value function $\nu : [0,1]^d \to \mathbb{R}$ that measures *utility* of subsets of players. Formally, the Shapley value for player j in game ν is defined as:

$$\phi_j(\nu) = \sum_{S \subseteq \Omega \setminus \{j\}} \left(|S|!(d - |S| - 1)!/d!\right)\left(\nu(S \cup j) - \nu(S)\right), \tag{3}$$

where $\Omega = \{1, ..., d\}$ is the set of players of the game. Given a value function ν, the Shapley values are proven to be the only credit allocation scheme that satisfies a particular set of favourable and fair game theoretical axioms, commonly known as *efficiency*, *null player property*, *symmetry* and *additivity* [21]. Štrumbelj and Kononenko [29] later connect Shapley values to the field of *explainable machine learning* by drawing an analogy between model fitting and cooperative game. Given a specific data point, by considering its *features* as *players* participating in a game that measures features' utilities, the Shapley values obtained can be treated as *local feature importance scores*. Such games are typically defined through the value functions defined below.

Definition 2.1 (Value functions)**.** *Let X be a random variable on $\mathcal{X} \subseteq \mathbb{R}^d$ and $f : \mathcal{X} \to \mathbb{R}$ a model from hypothesis space $\mathcal{H}$. The value function $\nu : \mathcal{X} \times [0,1]^d \times \mathcal{H}$ is given by*

$$\nu_{\mathbf{x},S}(f) = \mathbb{E}_{r(X_{S^c} \mid X_S = \mathbf{x}_S)}\left[f(\{X_S, X_{S^c}\}) \mid X_S = \mathbf{x}_S\right] \tag{4}$$

where r is an appropriate reference distribution, X_S is the subvector of X corresponding to the feature set S, S^c is the complement of the feature set S and $\{X_S, X_{S^c}\} = X$ denotes the concatenation of X_S and X_{S^c}.

In other words, given a data point $\mathbf{x}$, the utility of the feature subset S is defined as the impact on the model prediction, after "removing" the contribution from S^c via integration with respect to the reference distribution r. These "removal-based" strategies are common in the explainability literature [30]. Nonetheless, the correct choice of the reference distribution has been a long-standing debate [31]. Janzing et al. [32] argued from a causality perspective that the feature marginal distribution should be used as the reference distribution, i.e. $r(X_{S^c} \mid X_S = x_S) = p(X_{S^c})$ where p is the data distribution. On the other hand, Frye et al. [33] disagreed by pointing out these "marginal" value functions ignore feature correlations and lead to unintelligible explanations in higher-dimensional data, and they instead advocate the use of conditional distribution as reference, i.e. $r(X_{S^c} \mid X_S = x_S) = p(X_{S^c} \mid X_S = x_S)$. Thus, there is no consensus and in fact, Chen et al. [31] took a neutral stand and argued the choice depends on the application at hand. This also leads to design of value functions for specific problems, e.g. improving local estimation [34], incorporating causal knowledge [35, 36] and modelling structured data [37]. In this paper, we will design an appropriate value function for preference learning and show that naive application of the existing value function to preference learning will lead to unintuitive results.

Shapley value estimation. Given a data point $\mathbf{x}$ and a model f, estimating Shapley values consist of two main steps: Firstly, for each feature subset $S \subseteq \Omega$, estimate the value function $\nu_{\mathbf{x},f}(S)$ either by Monte Carlo sampling from the reference distributions r, or by utilising a model specific structure to speed up the estimation such as in LINEARSHAP [29], DEEPSHAP [14], TREESHAP [38], and RKHS-SHAP [12]. The former sampling procedure is straightforward when r is the marginal distribution, but computationally heavy and difficult when r is the conditional distribution, as it involves estimating an exponential number of conditional densities [39]. Finally, after estimating the value functions, one can compute the Shapley values based on Eq. 3 or by utilising the efficient weighted least square approach proposed by Lundberg and Lee [14].

Estimating value functions when $f \in \mathcal{H}_k$. We give a review to the recently introduced RKHS-SHAP algorithm proposed by Chau et al. [12] as it is another core component for PREF-SHAP. RKHS-SHAP is a SV estimation method for functions in a given RKHS. It circumvents the need for any density estimation and utilises the arsenal of kernel mean embeddings [40] to estimate the value functions non-parametrically. Assume k takes a product kernel structure across dimensions, then for any $f \in \mathcal{H}_k$, by applying the *reproducing property* [25], the value function can be decomposed as:

$$\nu_{\mathbf{x},S}(f) = \left\langle f,\ \mathbb{E}_{r(X_{S^c}|X_S=\mathbf{x}_S)}\left[k\left(\{X_S, X_{S^c}\}, \cdot\right) \mid X_S = \mathbf{x}_S\right]\right\rangle_{\mathcal{H}_k} \tag{5}$$

$$= \left\langle f, k_{X_S} \otimes \mu_{r(X_{S^c}|X_S=\mathbf{x}_S)}\right\rangle_{\mathcal{H}_k}, \tag{6}$$

where k_{X_S} is the product of kernels belonging to the feature set S, and $\mu_{r(X_{S^c}|X_S=\mathbf{x}_S)} := \int k_{X_{S^c}} r(X_{S^c} \mid X_S = \mathbf{x}_S) dX_{S^c}$ is the kernel mean embedding [40] of the reference distribution r. Depending on the choice of the reference distribution, one recovers either the standard kernel mean embedding or the conditional mean embedding. This allows us to arrive at a closed form expression of the value function and circumvents the need for fitting an exponential number of conditional densities.

3 Proposed method: PREF-SHAP

In this section, we will present PREF-SHAP, a new Shapley explainability toolbox designed to explain preferences by attributing contribution scores over item-level and context-level covariates for our preference models. Recall the likelihood model for C-GPM from Sec. 2.1:

$$p\left(y \mid \mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right) = \sigma\left(y g_U\left(\mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right)\right), \tag{7}$$

where g_U is the context-included preference function that denotes the strength of preference of item $\mathbf{x}^{(\ell)}$ over item $\mathbf{x}^{(r)}$ under context $\mathbf{u}$. As there are two distinct sets of covariates present, we will propose two different value functions to capture the influences from items and context variables respectively, and show how they could be estimated non-parametrically using tools from the kernel methods literature, as in RKHS-SHAP.

3.1 Preferential value function for items

To explain a general preference model $g : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, we propose the following *preferential value function for items*.

Definition 3.1 (Preferential value function for items)**.** *Given a preference function $g \in \mathcal{H}$, a pair of items $\left(\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right) \in \mathcal{X} \times \mathcal{X}$ to compare, we define the preferential value function for items as $\nu^{(p_I)} : \mathcal{X} \times \mathcal{X} \times [0,1]^d \times \mathcal{H} \to \mathbb{R}$ such that:*

$$\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}(g) = \mathbb{E}_q \left[g(\{X_S^{(\ell)}, X_{S^c}^{(\ell)}\}, \{X_S^{(r)}, X_{S^c}^{(r)}\}) \mid X_S^{(\ell)} = \mathbf{x}_S^{(\ell)}, X_S^{(r)} = \mathbf{x}_S^{(r)} \right] \tag{8}$$

where expectation is taken over the reference $q\left(X_{S^c}^{(\ell)}, X_{S^c}^{(r)} \mid X_S^{(\ell)} = \mathbf{x}_S^{(\ell)}, X_S^{(r)} = \mathbf{x}_S^{(r)}\right)$.

We note that $\nu^{(p_I)}$ is also applicable to the context-specific preference models. For example, applying $\nu^{(p_I)}$ to $g_{\mathbf{u}} := g_U(\mathbf{u}, \cdot, \cdot)$ allows one to quantify the item covariate's influences under a specific context $\mathbf{u}$, while applying $\nu^{(p_I)}$ to $\bar{g} := \mathbb{E}_{p(U)}[g_U(U, \cdot, \cdot)]$ quantifies the average influence from each of the item covariates instead.

Similar to standard value functions, the influence of a feature set S shared by the items $\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}$ is measured as the impact on the preference model after "removing" contributions from features in S^c, via integration with respect to some reference distribution r. Similar to g, this value function is skew-symmetric in its first two arguments, i.e. $\nu^{(p_I)}(\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}, S, g) = -\nu^{(p_I)}(\mathbf{x}^{(r)}, \mathbf{x}^{(\ell)}, S, g)$. This is justified, since features that "encourage" preference of $\mathbf{x}^{(\ell)}$ over $\mathbf{x}^{(r)}$ should naturally be the ones that "discourage" preference of $\mathbf{x}^{(r)}$ over $\mathbf{x}^{(\ell)}$ to ensure consistency. In this paper, we assume the items are i.i.d sampled from some distribution p, and we utilise the observational data distribution as reference as in [33], i.e. we take $r\left(X_{S^c}^{(\ell)}, X_{S^c}^{(r)} \mid X_S^{(\ell)} = \mathbf{x}_S^{(\ell)}, X_S^{(r)} = \mathbf{x}_S^{(r)}\right)$ to be $p\left(X_{S^c}^{(\ell)} \mid X_S^{(\ell)} = \mathbf{x}_S^{(\ell)}\right) p\left(X_{S^c}^{(r)} \mid X_S^{(r)} = \mathbf{x}_S^{(r)}\right)$. Although we decide here to use the observational distribution as the reference, the corresponding estimation procedure follows analogously if one instead uses the marginal distribution approach in Janzing et al. [32].

Problems with direct application of SHAP to preference model g A naive way of explaining with SHAP a general preference model g which assumes no rankability would require concatenation of the items' covariates. Namely, we would set $\mathbf{z} = (\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}) \in \mathbb{R}^{2d}$ and then apply SHAP to the function $g(\mathbf{z})$ directly, now giving $2d$ Shapley values for each observed preference, i.e. two Shapley values for each feature. Not only does this approach require us to consider a larger number of feature coalitions during computation (squaring the original amount), but it also ignores that $\mathbf{x}^{(\ell)}$ and $\mathbf{x}^{(r)}$ in fact consist of the same features, leading to inconsistent explanations, i.e. that the same feature in $\mathbf{x}^{(\ell)}$ and $\mathbf{x}^{(r)}$ has a different influence, hence giving different explanations simply due to the ordering of items. We illustrate these pitfalls of such a naive approach in Appendix B.

Empirical estimation of the preferential value function $\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}(g)$ While the *preferential value function* is general in the sense that it could be applied to any preference function g, we divert our attention to functions in $\mathcal{H}_{k_E}$, where k_E is the *generalised preferential kernel* introduced in Sec 2.1. This allows us to adapt the recently introduced RKHS-SHAP to our settings, and we can thus circumvent learning an exponential number of conditional densities as in [33]. In the following segment, we prove the existence of the Riesz representation of the *preferential value functional*, a necessary step to adapt the RKHS-SHAP framework to our setting.

Proposition 3.1 (Preferential value functional for items)**.** *Let k be a product kernel on $\mathcal{X}$, i.e. $k(\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}) = \prod_{j=1}^d k^{(j)}(x^{(j)}, x'^{(j)})$. Assume $k^{(j)}$ are bounded for all j, then the Riesz representation of the functional $\nu^{(p)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}$ exists and takes the form:*

$$\nu^{(p)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S} = \frac{1}{\sqrt{2}} \left(\mathcal{K}(\mathbf{x}^{(\ell)}, S) \otimes \mathcal{K}(\mathbf{x}^{(r)}, S) - \mathcal{K}(\mathbf{x}^{(r)}, S) \otimes \mathcal{K}(\mathbf{x}^{(\ell)}, S) \right)$$

where $\mathcal{K}(\mathbf{x}, S) = k_S(\cdot, \mathbf{x}_S) \otimes \mu_{X_{S^c}|X_S=\mathbf{x}_S}$ and $k_S(\cdot, \mathbf{x}_S) = \bigotimes_{j \in S} k^{(j)}(\cdot, x^{(j)})$ is the sub-product kernel defined analogously as X_S.

All proofs are included in the appendix. By representing the functionals as elements in the corresponding RKHS, we can now estimate the value function non-parametrically using kernel mean embeddings.

Proposition 3.2 (Non-parametric Estimation)**.** *Given $\hat{g} = \sum_{j=1}^m \alpha_j k_E((\mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot)$, datasets $\mathbf{X}^{(\ell)}, \mathbf{X}^{(r)}$, test items $\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}$, the preferential value function at test items $\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}$ for coalition S*

Table 1: A summary of how our preference value functions can tackle different explanation tasks

Candidate	Explanation of interest	Value function	Preference function
$\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}$	Which item features contributed most to this duel?	$\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}$	$g, \mathbb{E}_U[g_U(U,\cdot,\cdot)]$
$\mathbf{x}^{(\ell)}$	Which item features contributed most to $\mathbf{x}^{(\ell)}$'s matches?	$\frac{1}{n}\sum_{i=1}^{n}\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}_i,S}$	$g, \mathbb{E}_U[g_U(U,\cdot,\cdot)]$
$\mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}$	Which context features contributed most to this duel?	$\nu^{(p_U)}_{\mathbf{u},\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}$	g_U
$\mathbf{u}$	Which context features contributed most on average?	$\frac{1}{m}\sum_{j=1}^{m}\nu^{(p_U)}_{\mathbf{u},\mathbf{x}_j^{(\ell)},\mathbf{x}_j^{(r)},S'}$	g_U

and preference function $\hat{g}$ can be estimated as

$$\hat{\nu}^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}(\hat{g}) = \boldsymbol{\alpha}^\top \left(\Gamma(\mathbf{X}_S^{(\ell)}, \mathbf{x}_S^{(\ell)}) \odot \Gamma(\mathbf{X}_S^{(r)}, \mathbf{x}_S^{(r)}) - \Gamma(\mathbf{X}_S^{(\ell)}, \mathbf{x}_S^{(r)}) \odot \Gamma(\mathbf{X}_S^{(r)}, \mathbf{x}_S^{(\ell)})\right),$$

where $\Gamma(\mathbf{X}_S^{(\ell)}, \mathbf{x}_S^{(\ell)}) = \mathbf{K}_{\mathbf{X}_S^{(\ell)},\mathbf{x}_S^{(\ell)}} \odot \mathbf{K}_{\mathbf{X}_{S^c}^{(\ell)},\mathbf{X}_{S^c}} \mathbf{K}^{-1}_{\mathbf{X}_S,\lambda} \mathbf{K}_{\mathbf{X}_S,\mathbf{x}_S^{(\ell)}}, \mathbf{K}_{\mathbf{X}_S,\lambda} = \mathbf{K}_{\mathbf{X}_S,\mathbf{X}_S} + n\lambda I$, $\boldsymbol{\alpha} = \{\alpha_j\}_{j=1}^m$ *and* $\lambda > 0$ *is a regularisation parameter.*

3.2 Preferential value function for contexts

The influence an individual context feature in U has on a C-GPM function g_U can be measured by the following value function.

Proposition 3.3 (Preferential value function for contexts)**.** *Given a preference function $g_U \in \mathcal{H}_{k_E^U}$, denote $\Omega' = \{1, ..., d'\}$, then the utility of context features $S' \subseteq \Omega'$ on $\{\mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\}$ is measured by $\nu^{(p_U)}_{\mathbf{u},\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S'}(g_U) = \mathbb{E}[g_U\left(\{\mathbf{u}_S, U_{S^c}\}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right) \mid U_S = \mathbf{u}_S]$ where the expectation is taken over the observational distribution of U. Now, given a test triplet $(\mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)})$, if $\hat{g}_U = \sum_{j=1}^m \alpha_j k_E^U\left((\mathbf{u}_j, \mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot\right)$, the non-parametric estimator is:*

$$\hat{\nu}^{(p_U)}_{\mathbf{u},\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S'}(\hat{g}_U) = \boldsymbol{\alpha}^\top \left(\left(\mathbf{K}_{\mathbf{U}_{S'},\mathbf{u}_{S'}} \odot \mathbf{K}_{\mathbf{U}_{S'^c},\mathbf{U}_{S'^c}} \left(\mathbf{K}_{\mathbf{U}_{S'},\mathbf{U}_{S'}} + m\lambda' I\right)^{-1} \mathbf{K}_{\mathbf{U}_{S'},\mathbf{u}_{S'}}\right) \odot \Xi_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)}}\right)$$

where $\Xi_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)}} = \left(\mathbf{K}_{\mathbf{X}^{(\ell)},\mathbf{x}^{(\ell)}} \odot \mathbf{K}_{\mathbf{X}^{(r)},\mathbf{x}^{(r)}} - \mathbf{K}_{\mathbf{X}^{(r)},\mathbf{x}^{(\ell)}} \odot \mathbf{K}_{\mathbf{X}^{(\ell)},\mathbf{x}^{(r)}}\right)$.

Analogously, the average influence of a specific context feature can be computed by taking an average over all pairs of matches, i.e. by using a modified value function $\frac{1}{m}\sum_{j=1}^m \nu^{(p_U)}_{\mathbf{u},\mathbf{x}_j^{(\ell)},\mathbf{x}_j^{(r)},S'}(\hat{g}_U)$. We summarise different ways to modify the proposed preferential value functions to interrogate the preference models in Table 1.

Computational complexity of PREF-SHAP When computing GPM, it is fundamentally a *kernel ridge regression* (KRR), which naïvely has complexity $\mathcal{O}(n^3)$. There exists a multitude of approximation techniques for KRR, the most common type being the Nyström approximation [41]. For all our experiments, we use FALKON [42], a large-scale library for solving kernel logistic regression using preconditioned conjugate gradient descent and Nyström approximations. FALKON has a computational complexity of $\mathcal{O}(n\sqrt{n})$, which effectively becomes the complexity for GPM when using FALKON. As the value function for GPM requires estimating conditional mean embeddings, which in turn also are KRR's, one can appeal to FALKON again to reduce complexity to $\mathcal{O}(n\sqrt{n})$. We summarize the procedure of PREF-SHAP in Algorithm 1. We further detail computational details pertaining to computing coalitions S and batched conjugate gradient descent (`BatchedCGD`) in Appendix A.

4 Experiments

The main focus of our experiments is to illustrate the difference between explaining GPM (PREF-SHAP) and applying SHAP to UPM, thus highlighting the difference in explaining the mechanism of eliciting preferences and explaining the utility. When we explain UPM, we first explain how items $\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}$ affect their utilities $f(\mathbf{x}^{(\ell)}), f(\mathbf{x}^{(r)})$. Explaining the utility corresponds to calculating the value functions of the utilities $\nu_{\mathbf{x}^{(\ell)},S}(f)$ and $\nu_{\mathbf{x}^{(r)},S}(f)$. By linearity of SHAP values [14] and the simple structure relating preference and utilities in UPM, we can explain UPM by subtracting the Shapley values of $\mathbf{x}^{(\ell)}$ with $\mathbf{x}^{(r)}$. However, this type of explanation is only correct when data is rankable, which seldom happens in practice, thus motivating PREF-SHAP.

Algorithm 1 PREF-SHAP

Input: Solution $\boldsymbol{\alpha}$, datasets $\mathbf{X}^{(\ell)}, \mathbf{X}^{(r)}, \mathbf{X}, \mathbf{U}$, test items $\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}, \mathbf{u}$, batch size n_b, number of coalition samples n_S, context-specific flag `cflg`

Compute effective dimension d_{eff} := Number of features with variance greater than 0.
Compute coalitions $\mathcal{S} = \{S_1, \ldots, S_{n_S}\}$, form binary matrix $\mathbf{Z} \in \{0,1\}^{n_S, d_{\text{eff}}}$ from $\mathcal{S}$, and compute weights $\mathbf{W} = [w_1, \ldots, w_{n_S}]$ with $w_i = \frac{d-1}{\binom{d}{|S_i|}|S_i|(d-|S_i|)}$.
for batch $\mathcal{S}_b$ in $\mathcal{S}$ **do**
    if `cflg` Take S, S^c of $\mathbf{X}^{(\ell)}, \mathbf{X}^{(r)}, \mathbf{X}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}$ **else** Take S', S'^c of $\mathbf{U}, \mathbf{u}$
    if `cflg` Compute $\mathbf{K}^{-1}_{\mathbf{X}_S,\lambda}[\mathbf{K}_{\mathbf{X}_S,\mathbf{x}^{(\ell)}_S}, \mathbf{K}_{\mathbf{X}_S,\mathbf{x}^{(r)}_S}]$ **else** $\left(\mathbf{K}_{\mathbf{U}_{S'},\mathbf{U}_{S'}} + m\lambda' I\right)^{-1} \mathbf{K}_{\mathbf{U}_{S'},\mathbf{u}_{S'}}$ using `BatchedCGD`
    if `cflg` Compute $\hat{\nu}^{(p)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},\mathcal{S}_b}(\hat{g})$ **else** $\hat{\nu}_{\mathbf{u},\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},\mathcal{S}'_b}(\hat{g}_U)$
end for
if `cflg` Set $\mathbf{v}_x = \{\hat{\nu}^{(p)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},\mathcal{S}_b}(\hat{g})\}_{b=1}^B$ **else** $\mathbf{v}_x = \{\hat{\nu}_{\mathbf{u},\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},\mathcal{S}'_b}(\hat{g}_U)\}_{b=1}^B$
Calculate Shapley values $\boldsymbol{\beta}_x = \left(\mathbf{Z}^\top \mathbf{W} \mathbf{Z}\right)^{-1} \mathbf{Z}^\top \mathbf{W} \mathbf{v}_x$
return $\boldsymbol{\beta}_x$

We apply PREF-SHAP to unrankable synthetic and real-world datasets to connect theory with practice. We split data, i.e. matches with their outcomes, into train (80%), validation (10%), and test (10%) and explain the model on a random subset of the data. The hyperparameters for the kernels are selected using gradient descent, based on the proposed method in [43]. We first generate synthetic duelling data where performance can be compared against ground truth, to demonstrate that PREF-SHAP is capable of identifying the relevant features.

Synthetic data We first consider a synthetic experiment with unrankable duelling data. We generate the items by first sampling 1000 item covariates $[x_i^{[0]}, x_i^{AB}, x_i^{AC}, x_i^{BC}] =: \mathbf{p}_i \in \mathbb{R}^4 \sim \mathcal{N}(0, \mathbf{I}_4)$. We associate each item with a cluster membership $c_i \in \{A, B, C\}$, where the assignment is randomly chosen for each item with equal probability. We then form the full item covariate by concatenating $\mathbf{p}_i$ with one-hot encoded c_i as $\mathbf{x}_i = [\mathbf{p}_i, \texttt{one_hot}(c_i)]$. 40000 matches between randomly chosen pairs of items are conducted by the following mechanism: match outcomes are decided based on the underlying cluster membership of the items. For example, if an item from cluster A competes against an item from cluster B, the winner is decided by their inter-cluster covariate x^{AB}, i.e. $i \preceq j$ if $x_j^{AB} \geq x_i^{AB}$. When the match is between members of the same cluster, it is dictated by the maximum among the within-cluster variable, i.e. $\max(x_i^{[0]}, x_j^{[0]})$. See Fig. 1 for an illustration. As no clusters have any advantage over the others, the data is not rankable, and we expect the inter-cluster covariates x^{AB}, x^{AC}, x^{BC} to have similar explanations on average, but significantly different from each other when we examine local explanations.

We consider both global and *grouped-local* explanations of the synthetic dataset in Figure 2 and Figure 3 respectively. In the global explanations, we explain all matches regardless of the cluster membership, while in the grouped-local explanations we only explain matches between items from A against items from B. For more grouped-local explanations on different cluster pairs, we refer to appendix B.

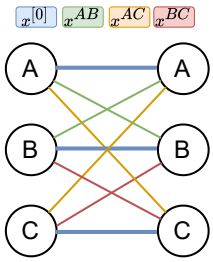

Figure 1: An illustration of our simulation: each edge corresponds to the variable that dictates the comparison based on the colour.

Interpreting the simulation explanations. The beehive plots showcase the recovered PREF-SHAP values, where the bar plots demonstrated the average PREF-SHAP values for each feature. The colour in the beehive plots indicates the magnitude of the difference between the corresponding features of the winner and of the loser in that match. For example, a red point in a beehive plot for feature d indicates that the difference $x^{(d)}_{winner} - x^{(d)}_{loser}$ is large.

Fig. 2 illustrates the explanation results for the global synthetic experiments. We see that PREF-SHAP identified the within-cluster variable $x^{[0]}$ as the most important, which is a consequence of the fact that the largest number of matches are played between the items of the same cluster (cf. Fig. 1 where there are three blue lines and two lines of each of the other colours). The three inter-cluster

Table 2: Dataset summary

Dataset	N_{Matches}	N_{items}	N_{Context}	$D_{\text{continuous}}$	D_{binary}	$D^{\text{Context}}_{\text{continuous}}$	$D^{\text{Context}}_{\text{binary}}$
Synthetic	40000	1000	-	4	3	-	-
Chameleon	106	35	-	7	19	-	-
Pokémon	60000	800	-	7	0	-	-
Tennis	95359	3483	4114 (tournaments)	4	7	0	6

variables contributed similarly according to PREF-SHAP, which, by symmetry, should be the case. Furthermore, the correct battle mechanism is captured by PREF-SHAP but not UPM, as we see that the large PREF-SHAP values for each feature are red in the beehive plot. This indicates that items with larger value are more likely to win against items with lower value in the corresponding features. In contrast, SHAP for UPM does not recover this insight.

The explanations for the matches between items from A against items from B, are shown in Fig 3. Here, x^{AB} is correctly picked as the relevant feature in these matches with PREF-SHAP, but not with SHAP for UPM. We see again that there is a clear tendency that large PREF-SHAP values are red for feature x^{AB}, showing that PREF-SHAP once again captures the designed gaming mechanism – which is not the case in SHAP for UPM. Intuitively, even though SHAP for UPM allows local explanations, it does so based on a *global utility*, which fails completely in a non-rankable case.

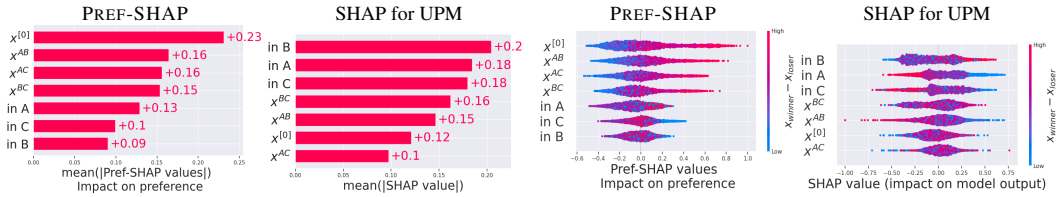

Figure 2: Bar and Beehive plots for global explanations on the synthetic dataset.

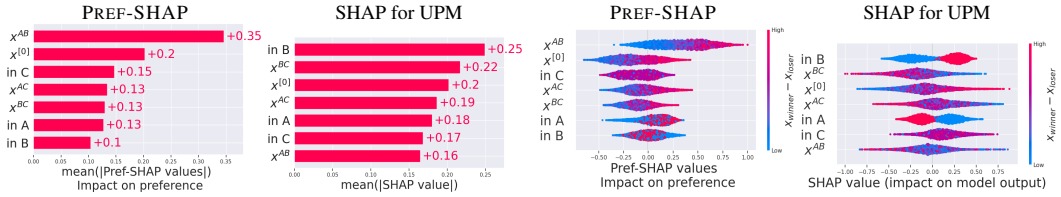

Figure 3: Bar and Beehive plots for grouped local explanations on the synthetic dataset (Cluster A vs B).

Real-world explanations For our real-world datasets, we consider publicly available datasets *Chameleon*, *Pokémon* and *Tennis*. We provide descriptive statistics of these datasets in Table 5 and give their brief descriptions below. Appendix B contains further large scale experiments on an additional dataset consisting of user-item interactions on a fashion retail website.

The Chameleon dataset [44] considers 106 contests between 35 male dwarf chameleons. Physical traits of the chameleons are measured such as the *height of their casque*, *length of their jaw*, *body mass* etc. According to [44], they fitted a linear Bradley Terry model and examined the coefficients to deduce that *casque height* (ch.res) and *relative area of the flank patch* (prop.patch) positively affected the fighting ability the most. *The Pokémon* dataset considers 60000 Pokémon battles among 800 Pokémon. Pokémon have different characteristics such as *attack power*, *speed*, *health* etc. The Pokémon further has at least one different *type* such as *Electric, Water, Fire*, etc. Certain types have advantages and disadvantages against each other, for instance, fire Pokémon are weak to water-based attacks (receiving twice the damage) and as a result have a disadvantage against water Pokémon.

The Tennis dataset considers professional tennis matches between 1991 and 2017 in all major tournaments each year. The data is provided publicly by ATP World Tour [45]. Features such as *birthyear*, *weight*, *height* etc are included about each tennis player together with context details of the match such as the court being indoor or outdoor and what surface the match is being played on.

The above datasets are not rankable, and we validate this claim by comparing GPM performance against UPM in Table 4, together with the estimated rankability measure *SpecR* proposed in [23] for each dataset. *SpecR* measures the similarity of the data to a complete dominance graph (i.e. rankable data). It takes values between 0 and 1 with values close to 1 being evidence in support of rankability.

Table 3: GPM vs UPM. Mean and standard deviations of performance averaged over 5 runs.

	Synthetic		Chameleon		Pokémon		Tennis	
	GPM	UPM	GPM	UPM	GPM	UPM	C-GPM	UPM
Test AUC	$0.98_{\pm 0.00}$	$0.71_{\pm}0.01$	$0.92_{\pm 0.07}$	$0.80_{\pm 0.07}$	$0.86_{\pm 0.00}$	$0.82_{\pm 0.00}$	$0.58_{\pm 0.02}$	$0.52_{\pm 0.02}$
SpecR	0.09		0.24		0.20		$0.13_{\pm 0.07}$	

For the Tennis data where there are additional relationships with the context (tournaments), we estimate the average *SpecR* of each tournament. Both the superior performance of GPM over UPM and the low *SpecR* measures suggest that the datasets are generally not rankable, which points to limitations of explaining preferences via utility-based modeling.

Explaining Pokémon battles. We first consider standard dueling data for explaining preferences. We explain the learned preferences and learned differences in utilities on the Pokémon dataset in Figure 4. In this dataset, we have summed the Shapley values for *Type* features.

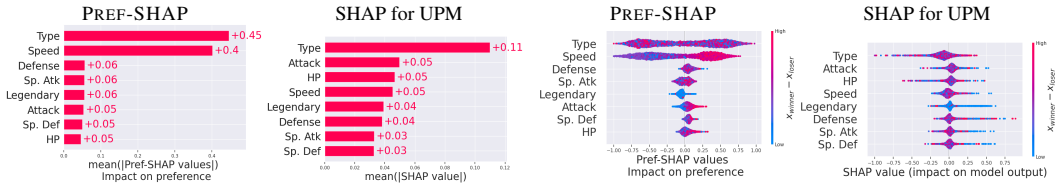

Figure 4: Bar and Beehive plots for the Pokémon dataset. PREF-SHAP captures that both speed and type matter, while SHAP for UPM only captures the type importance.

We see that explaining general preferences provides further insight than just explaining the difference in utility functions. In particular, SHAP for UPM does not capture the additional importance of *Speed* in winning battles. As higher (more red) values of differences in speed $x^{\text{speed}}_{\text{winner}} - x^{\text{speed}}_{\text{loser}}$ have positive impact on the outcome, we conclude that having higher speed than your opponent is advantageous besides a type advantage. This insight is aligned with the "Sweeper" strategy [46], where one would employ a leading Pokémon with very high speed and attack to attempt downing the opponent before they can strike back. In Fig. 5, we see PREF-SHAP can also capture the correct type advantage/disadvantages among the Pokémon, but not SHAP for UPM.

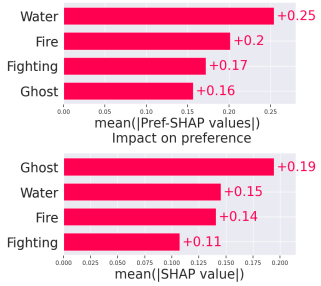

Figure 5: Explaining matches between 4 types of Pokémon, among them only fire and water has a type disadvantage/advantage against each other. PREF-SHAP (top) correctly identifies that fire and water are the most important, while water and fire are not deemed most important by SHAP for UPM.

Explaining Chameleon contests. We find that UPM's explanations are more aligned with [44]'s findings (*prop.path* and *ch.res* are the most important features), which is unsurprising since the Bradley Terry model used in [44] is also a utility based model. However, since GPM gives a much better predictive performance than UPM (Test AUC 0.92 v.s. 0.80), we believe PREF-SHAP's explanations are also insightful. In fact, PREF-SHAP discovers that having larger *jaw sizes* (jl.res) than your opponent have a significant negative effect on match outcome, a previously undiscovered mechanism from [44]. We verify this finding in Appendix B by applying PREF-SHAP to GPM trained on multiple folds of the Chameleon dataset and consistently find that high values of the *jaw size* (jl.res) variable have a negative impact on the outcome.

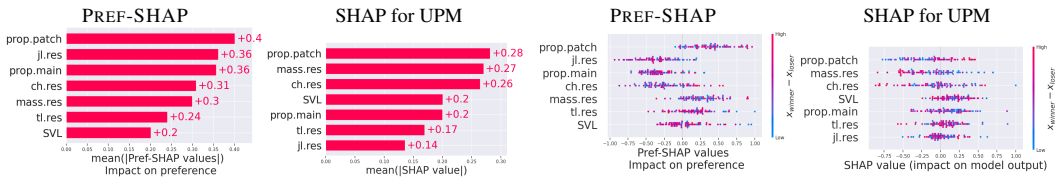

Figure 6: Bar and Beehive plots for the Chameleon dataset

Explaining Tennis matches. We now consider preference learning with context covariates and explain both item characteristics and context covariates in Figure 7. In terms of item-based inference, PREF-SHAP finds that being older than your opponent ($x^{\text{yob}}_{\text{winner}} - x^{\text{yob}}_{\text{loser}} < 0 \rightarrow$ Blue), physically

heavier, and taller than your opponent positively impacts the chances of winning. We also find that debuting earlier as a professional tennis player than your opponent positively impacts your chances of winning. This is not surprising as debuting earlier may be indicative of a promising young talent. Across all competitions, there appear to be no significant patterns in environment effects.

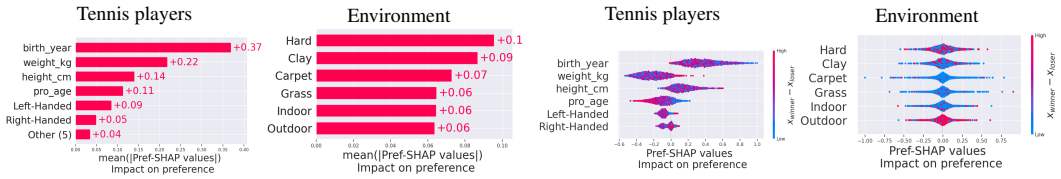

Figure 7: Item and context-specific Pref-SHAP values for the Tennis dataset

Explaining Djokovic's losses In plot Figure 8, we locally explain all Novak Djokovic's losses in his professional career. Novak Djokovic is regarded as one of the greatest tennis players of all time, so understanding his weakness could serve as a practical demonstration of the utility of PREF-SHAP.

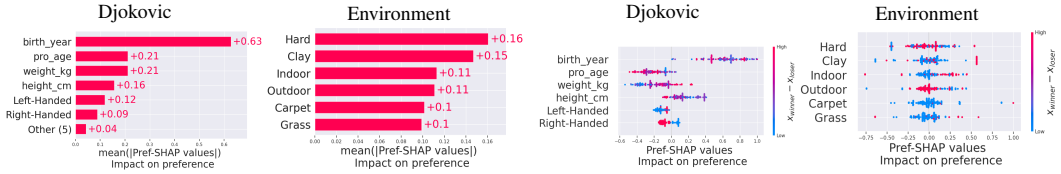

Figure 8: Local explanations of Djokovic losses

While the results take a similar shape to the global explanations, Djokovic remarkably seems to be weaker to players shorter than him, contrary to the general advantage of being taller. Besides this, Djokovic seems to be weaker on clay courts and when playing indoors.

5 Conclusion

In this work, we proposed PREF-SHAP to explain preference learning for pairwise comparison data. We proposed the appropriate value function for preference explanations and demonstrated the pathologies of the naive concatenation approach in Appendix B. Experiments demonstrated that PREF-SHAP recovers richer explanations than utility-based approaches, showcasing the ability of PREF-SHAP in interpreting the mechanism of preference elicitation.

Acknowledgments

The authors sincerely thank Oscar Clivio for the helpful comments and feedback. SLC is supported by the EPSRC and MRC through the OxWaSP CDT programme EP/L016710/1. DS is supported in part by Tencent AI Lab and in part by the Alan Turing Institute (EP/N510129/1).

References

[1] Johannes Fürnkranz and Eyke Hüllermeier. Pairwise preference learning and ranking. In Nada Lavrač, Dragan Gamberger, Hendrik Blockeel, and Ljupčo Todorovski, editors, *Machine Learning: ECML 2003*, pages 145–156, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-39857-8.

[2] Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[3] Louis L Thurstone. A law of comparative judgment. *Psychological review*, 101(2):266, 1994.

[4] Wei Chu and Zoubin Ghahramani. Preference learning with Gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 137–144, 2005.

[5] Javier González, Zhenwen Dai, Andreas Damianou, and Neil D Lawrence. Preferential Bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1282–1291, 2017.

[6] Manuela Cattelan, Cristiano Varin, and David Firth. Dynamic bradley–terry modelling of sports tournaments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 62(1): 135–150, 2013.

[7] Siu Lun Chau, Mihai Cucuringu, and Dino Sejdinovic. Spectral ranking with covariates. *arXiv preprint arXiv:2005.04035*, 2020.

[8] Daniel Kahneman and Amos Tversky. On the interpretation of intuitive probability: A reply to Jonathan Cohen. *Cognition*, 7(4):409–411, 1979.

[9] Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Jose M Hernández-Lobato. Collaborative gaussian processes for preference learning. In *Advances in neural information processing systems*, pages 2096–2104, 2012.

[10] Stefanos Bennett, Mihai Cucuringu, and Gesine Reinert. Lead-lag detection and network clustering for multivariate time series with an application to the us equity market. *arXiv preprint arXiv:2201.08283*, 2022.

[11] Devi M Stuart-Fox, David Firth, Adnan Moussalli, and Martin J Whiting. Multiple signals in chameleon contests: designing and analysing animal contests as a tournament. *Animal Behaviour*, 71(6):1263–1271, 2006.

[12] Siu Lun Chau, Javier Gonzalez, and Dino Sejdinovic. RKHS-SHAP: Shapley values for kernel methods. *arXiv preprint arXiv:2110.09167*, 2021.

[13] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.

[14] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

[15] Tapio Pahikkala, Willem Waegeman, Evgeni Tsivtsivadze, Tapio Salakoski, and Bernard De Baets. Learning intransitive reciprocal relations with kernel methods. *European Journal of Operational Research*, 206(3):676–685, 2010.

[16] Siu Lun Chau, Javier Gonzalez, and Dino Sejdinovic. Learning inconsistent preferences with gaussian processes. *International Conference on Artificial Intelligence and Statistics*, 2022.

[17] Christian List. Social Choice Theory. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2022 edition, 2022.

[18] William V Gehrlein. Condorcet's paradox. *Theory and Decision*, 15(2):161–197, 1983.

[19] Rosy Tsopra, Jean-Baptiste Lamy, and Karima Sedki. Using preference learning for detecting inconsistencies in clinical practice guidelines: Methods and application to antibiotherapy. *Artificial Intelligence in Medicine*, 89, 04 2018. doi: 10.1016/j.artmed.2018.04.013.

[20] Yifan Feng, René Caldentey, and Christopher Ryan. Robust learning of consumer preferences. *Operations Research*, 12 2021. doi: 10.1287/opre.2021.2157.

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

[22] Code for Pref-SHAP. https://github.com/MrHuff/PREF-SHAP.

[23] Paul Anderson, Timothy Chartier, and Amy Langville. The rankability of data. *SIAM Journal on Mathematics of Data Science*, 1(1):121–143, 2019.

[24] Thomas R Cameron, Amy N Langville, and Heather C Smith. On the graph laplacian and the rankability of data. *Linear Algebra and its Applications*, 588:81–100, 2020.

[25] Vern I Paulsen and Mrinal Raghupathi. *An introduction to the theory of reproducing kernel Hilbert spaces*, volume 152. Cambridge university press, 2016.

[26] Bharath K Sriperumbudur, Kenji Fukumizu, and Gert RG Lanckriet. Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research*, 12(Jul): 2389–2410, 2011.

[27] Paul Anand. Are the preference axioms really rational? *Theory and Decision*, 23(2):189–214, Sep 1987. ISSN 1573-7187. doi: 10.1007/BF00126305. URL `https://doi.org/10.1007/BF00126305`.

[28] Robert J. Aumann. Utility theory without the completeness axiom: A correction. *Econometrica*, 32(1/2):210–212, 1964. ISSN 00129682, 14680262. URL `http://www.jstor.org/stable/1913746`.

[29] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014.

[30] Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021.

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

[32] Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. Feature relevance quantification in explainable ai: A causal problem. In *International Conference on Artificial Intelligence and Statistics*, pages 2907–2916, 2020.

[33] Christopher Frye, Damien de Mijolla, Laurence Cowton, Megan Stanley, and Ilya Feige. Shapley-based explainability on the data manifold. *arXiv preprint arXiv:2006.01272*, 2020.

[34] Sahra Ghalebikesabi, Lucile Ter-Minassian, Karla DiazOrdaz, and Chris C Holmes. On locality of local explanation models. *Advances in Neural Information Processing Systems*, 34, 2021.

[35] Christopher Frye, Ilya Feige, and Colin Rowat. Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability. *arXiv preprint arXiv:1910.06358*, 2019.

[36] Tom Heskes, Evi Sijben, Ioan Gabriel Bucur, and Tom Claassen. Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. *Advances in neural information processing systems*, 33:4778–4789, 2020.

[37] Alexandre Duval and Fragkiskos D Malliaros. Graphsvx: Shapley value explanations for graph neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 302–318. Springer, 2021.

[38] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.

[39] Chih-Kuan Yeh, Kuan-Yun Lee, Frederick Liu, and Pradeep Ravikumar. Threading the needle of on and off-manifold value functions for shapley explanations. *arXiv preprint arXiv:2202.11919*, 2022.

[40] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel mean embedding of distributions: A review and beyond. *arXiv preprint arXiv:1605.09522*, 2016.

[41] Petros Drineas and Michael W. Mahoney. On the nystrom method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6(72): 2153–2175, 2005. URL `http://jmlr.org/papers/v6/drineas05a.html`.

[42] Giacomo Meanti, Luigi Carratino, Lorenzo Rosasco, and Alessandro Rudi. Kernel methods through the roof: handling billions of points efficiently. *ArXiv*, abs/2006.10350, 2020.

[43] Giacomo Meanti, Luigi Carratino, Ernesto De Vito, and Lorenzo Rosasco. Efficient hyperparameter tuning for large scale kernel ridge regression, 2022. URL `https://arxiv.org/abs/2201.06314`.

[44] Devi Stuart-Fox, David Firth, Adnan Moussalli, and Martin Whiting. Multiple signals in chameleon contests: Designing and analysing animal contests as a tournament. *Animal Behaviour*, 71:1263–1271, 06 2006. doi: 10.1016/j.anbehav.2005.07.028.

[45] Tennis dataset. `https://datahub.io/sports-data/atp-world-tour-tennis-data`, 2022.

[46] Sweeper Strategy Description. `https://strategywiki.org/wiki/Pokémon/Competitive_battling/The_"Job_System"`.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? See Appendix A [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Computation and Implementation Details

We propose several optimizations in the PREF-SHAP procedure. We consider fast sampling of coalitions S in Algorithm 2 batched conjugate gradient descent in Algorithm 3 described below.

Fast coalitions We first propose an optimized sampling scheme for finding coalitions S in Algorithm 2. In contrast to the implementation in [14] which samples the weights from $p(Z)$, our method

Algorithm 2 Sampling unique n, d-dimensional coalitions in $\mathcal{O}(d)$ time

Input: Number of coalitions n, number of features d.
$I = \text{SampleWithoutReplacement}(n, 0, 2^d)$ ▷ Sample n unique integers between 0 and 2^d
def base2(i): ▷ Convert integer to base-2 representation
  $S = [0, \dots, 0] \in \mathbb{R}^d$, $r = i$ ▷ Initialize d-dimensional 0 vector and the rest term r
  while $r > 0$ **do**
    $i = \lfloor \log_2(r) \rfloor$ ▷ Find which index of S to set to 1
    $S[i] = 1$ ▷ Update S
    $r = r - 2^i$ ▷ Update rest term
  end while
  return S
return $\{S_1, \dots S_n\} = \text{parallel_apply}(I, \text{base2})$ ▷ Each integer can be independently converted

is embarrassingly parallel, which allows for an additional $\mathcal{O}(n)$ reduction. A naive algorithm that compares each sample S_i has complexity $\mathcal{O}(n^2 d^2)$ and cannot be parallelized.

Stabilizing the Shapley value estimation We remove the features which have 0 variance in the data we are explaining, similar to the implementation in SHAP. To ensure we get numerically stable Shapley Values, we calculate the inverse using Cholesky decomposition, as we found the regular inverse function provided inconsistent results.

To calculate CMEs effectively, we use *preconditioned batched* conjugate gradient descent over coalitions detailed in Algorithm 3.

Algorithm 3 Batched conjugate gradient descent

Input: Preconditioner $P = \mathbf{K}_{\mathbf{x}_D}^{-1}$, batch $\mathbf{X} = [\mathbf{K}_{\mathbf{x}_{S_i}} \dots \mathbf{K}_{\mathbf{x}_{S_n}}]$, $\mathbf{B} = [\mathbf{K}_{\mathbf{x}_{S_i}, x_{S_i}} \dots \mathbf{K}_{\mathbf{x}_{S_n}, x_{S_n}}]$, `max_its`, tolerance ε
Set $\mathbf{R} = \mathbf{B}$, $\mathbf{Z} = \text{BatchMM}(P, \mathbf{B})$, $\mathbf{p} = \mathbf{Z}$, $\mathbf{a} = \mathbf{0}$
Set $\mathbf{R}_Z = [(\mathbf{R}_i \circ \mathbf{Z}_i)_{++} \dots (\mathbf{R}_n \circ \mathbf{Z}_n)_{++}]$ ▷ Element wise product and sum
for `max_its` **do**
  $\mathbf{L} = \text{BatchMM}(\mathbf{X}, \mathbf{B})$
  $\boldsymbol{\alpha} = \mathbf{L} \circ \frac{1}{[(\mathbf{L}_i \circ \mathbf{p}_i)_{++} \dots (\mathbf{L}_n \circ \mathbf{p}_n)_{++}]}$
  $\mathbf{a} = \mathbf{a} + \boldsymbol{\alpha} \circ \mathbf{p}$
  $\mathbf{r} = \mathbf{r} - \boldsymbol{\alpha} \circ \mathbf{L}$
  if $\text{Mean}([(\mathbf{r}_i \circ \mathbf{r}_i)_{++} \dots (\mathbf{r}_n \circ \mathbf{r}_n)_{++}]) < \varepsilon$ **then return** $\mathbf{a}$
  end if
  $\mathbf{z} = \text{BatchMM}(P, \mathbf{r})$
  $\mathbf{R}_Z^{\text{new}} = [(\mathbf{R}_i \circ \mathbf{Z}_i)_{++} \dots (\mathbf{R}_n \circ \mathbf{Z}_n)_{++}]$
  $\mathbf{p} = \mathbf{z} + [(\mathbf{R}_i^{\text{new}} \circ \frac{1}{\mathbf{R}_i})_{++} \dots (\mathbf{R}_n^{\text{new}} \circ \frac{1}{\mathbf{R}_n})_{++}] \circ \mathbf{p}$
  $\mathbf{R}_Z = \mathbf{R}_Z^{\text{new}}$
end for
return $\mathbf{a}$

We have run all our jobs on one Nvidia V100 GPU.

B Additional Experimental Results

Naive Concatenation We demonstrate the pathologies of the naive concatenation approach mentioned in Sec. 3 with our synthetic experiment. Recall that naive-concatenation approach here corresponds to first concatenating $\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}$'s features together and applying SHAP to the learned

function g directly, in order to to obtain $2d$ Shapley values, instead of the original d, since each feature has been duplicated. This approach ignores that the items $\mathbf{x}^{(\ell)}$ and $\mathbf{x}^{(r)}$ in fact consist of the same features. Therefore, when we use the usual value function from SHAP (corresponding to the impact an individual feature has on the model when it is turned "off" by integration), we would be turning "off" the feature from the left item, while keeping "on" the feature from the right item, obtaining a difficult to interpret attribution score. This is highly problematic, as we might be inferring vastly different contributions of the same feature purely because of the item ordering when concatenating them. We note that the item ordering in all our experiments is arbitrary and carries no additional information about the match.

We can see from Fig. 9 that when we explain the preference model applied to the synthetic experiment, we see that, for example, $x^{AC}(l)$ from $\mathbf{x}^{(\ell)}$ and $x^{AC}(r)$ from $\mathbf{x}^{(r)}$ have in fact very different average Shapley values. Even attempting to average each pair of corresponding features does not give a meaningful feature contribution ordering (x^{AB} and x^{AC} are scored higher on average than x^{BC} and $x^{[0]}$).

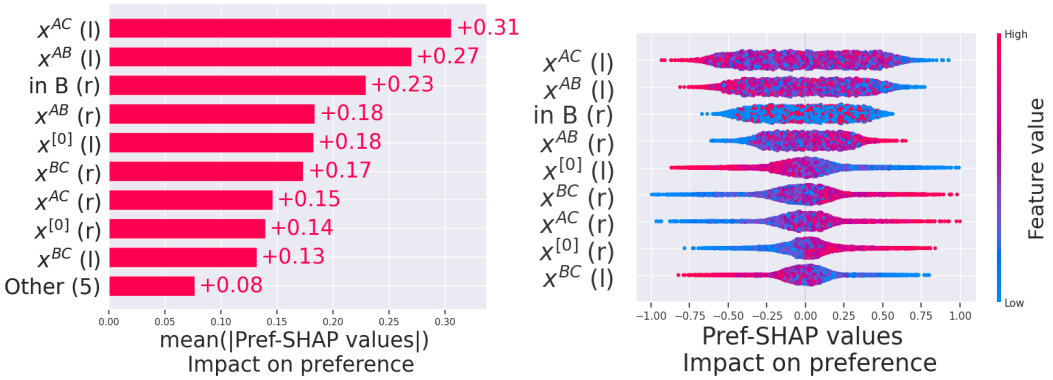

Figure 9: Explaining a naïve concatenation model

Additional synthetic data We consider an additional synthetic experiment where we generate data directly from a GPM model and one where we construct synthetic dueling data. When simulating data, we first generate player covariates as $\mathbf{x}_i \in \mathbb{R}^d \sim \mathcal{N}(0, 0.1\mathbf{I}_d)$ for each player i. When generating from the GPM model, we would set 2 covariates as important, by only keeping the 2 first entries of $\mathbf{x}_i$ and fixing the rest to be constant (equal to 0). We build a GPM model for g out of these covariates and generate match outcomes.

We consider $d = 10$, where only the two first features are set to be important in predicting the outcome.

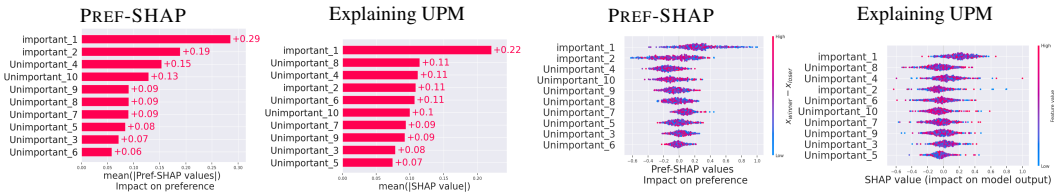

Figure 10: Bar and Beehive plots for Simulation A. PREF-SHAP recovers the correct features (1,2), while explaining UPM does not.

Chameleon data We further provide explanations of the Chameleon dataset on several different folds in Figure 11 and Figure 12.

Additional local explanations We additionally provide local explanations on the synthetic dataset in Figure 13 and Figure 14.

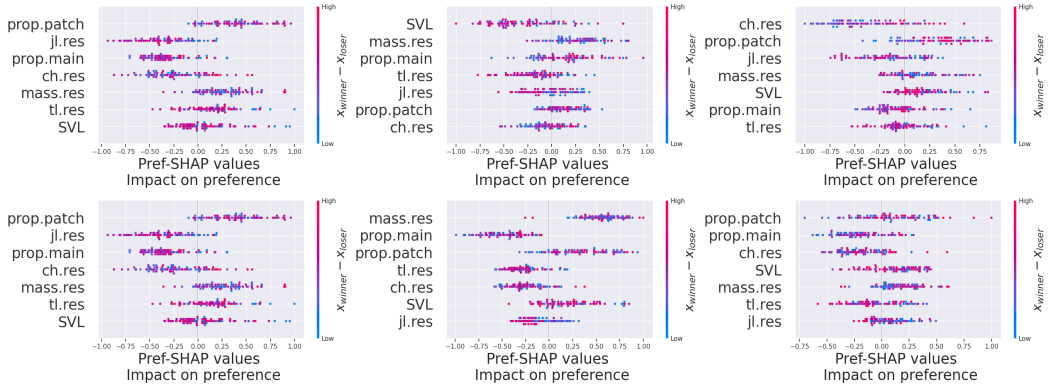

Figure 11: PREF-SHAP applied to 6 different folds of Chameleon. PREF-SHAP consistently finds that higher values of *jaw length* (jl.res) have a negative impact on outcome.

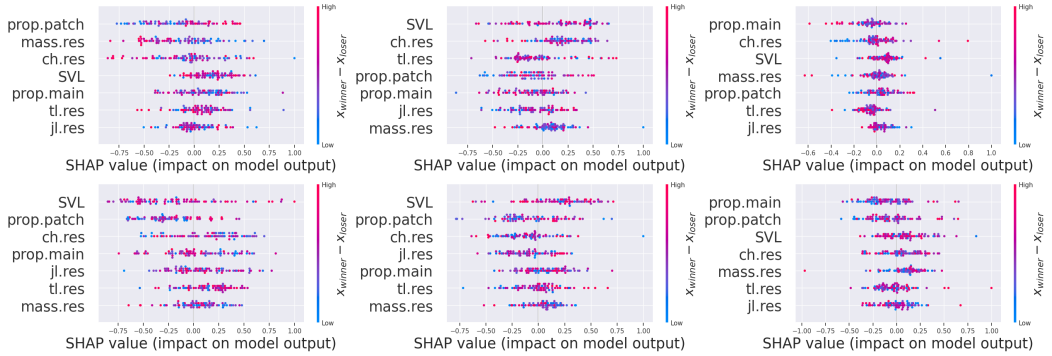

Figure 12: UPM explanations on 6 different folds of Chameleon. UPM is unable to find a consistent pattern for the impact of *jaw length* (jl.res) on the outcome.

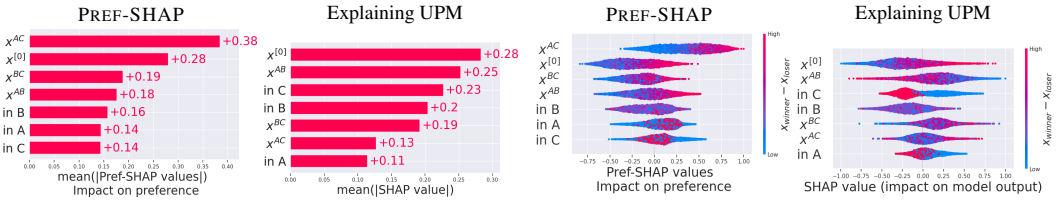

Figure 13: Explaining matches between clusters 0 and 2 on the synthetic dataset.

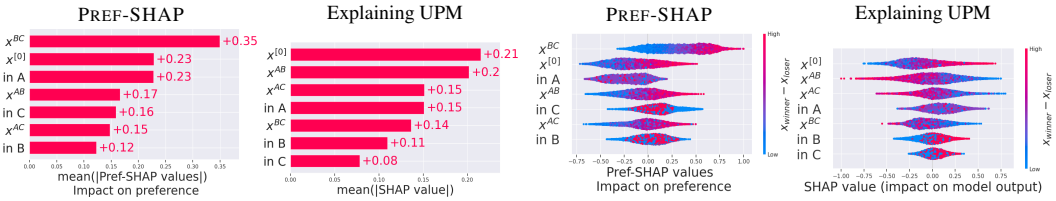

Figure 14: Explaining matches between clusters 1 and 2 on the synthetic dataset.

Website dataset *The Website* dataset considers anonymized visitors on a fashion retail website, where we are given what garment each visitor viewed and what each visitor clicked in a session. A user may have more than one session. In this setup, we interpret a browsing session for a visitor as multiple matches between items, such that the winning item (clicked) competes against all losing items (only viewed). If several items are winners, they do not play against each other. Each item has several descriptive statistics such as *colour*, *garment type*, *assortment characteristic* etc. There

Table 5: Dataset summary

Dataset	N_{Matches}	N_{Players}	N_{Context}	$D_{\text{continuous}}$	D_{binary}	$D^{\text{Context}}_{\text{continuous}}$	$D^{\text{Context}}_{\text{binary}}$
Website	85144	20626	129117 (users)	0	93	1	4

are some limited descriptive statistics of the visitors, such as *year of birth* and *gender code* (i.e. Male/Female/Unspecified/Unknown).

Explaining Website For the website dataset, we explain product and user preferences in Figure 15. We generally found that, for the period considered, cosmetic products and the "Jersey Basic category" drove clicks.

Figure 15: Barplots and beeplots for the website dataset, products on the left and user variables on the right.

Table 4: GPM vs UPM. Mean and standard deviations of performance averaged over 5 runs.

	Synthetic		Chameleon		Pokémon		Tennis		Website	
	GPM	UPM	GPM	UPM	GPM	UPM	C-GPM	UPM	C-GPM	UPM
Test AUC	$0.98_{\pm 0.00}$	$0.71_{\pm 0.01}$	$0.92_{\pm 0.07}$	$0.80_{\pm 0.07}$	$0.86_{\pm 0.00}$	$0.82_{\pm 0.00}$	$0.58_{\pm 0.02}$	$0.52_{\pm 0.02}$	$0.66_{\pm 0.01}$	$0.65_{\pm 0.01}$
SpecR	0.09		0.24		0.20		$0.13_{\pm 0.07}$		$0.53_{\pm 0.10}$	

C Proofs

Proposition 3.1 (Preferential value functional for items)**.** *Let k be a product kernel on $\mathcal{X}$, i.e. $k(\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}) = \prod_{j=1}^{d} k^{(j)}(x^{(j)}, x'^{(j)})$. Assume $k^{(j)}$ are bounded for all j, then the Riesz representation of the functional $\nu^{(p)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}$ exists and takes the form:*

$$\nu^{(p)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S} = \frac{1}{\sqrt{2}}\Big(\mathcal{K}(\mathbf{x}^{(\ell)}, S) \otimes \mathcal{K}(\mathbf{x}^{(r)}, S) - \mathcal{K}(\mathbf{x}^{(r)}, S) \otimes \mathcal{K}(\mathbf{x}^{(\ell)}, S)\Big)$$

where $\mathcal{K}(\mathbf{x}, S) = k_S(\cdot, \mathbf{x}_S) \otimes \mu_{X_{S^c}|X_S=\mathbf{x}_S}$ and $k_S(\cdot, \mathbf{x}_S) = \bigotimes_{j\in S} k^{(j)}(\cdot, x^{(j)})$ is the sub-product kernel defined analogously as X_S.

Proof. From [16], we know the generalised preferential kernel has the following feature map:

$$k_E((\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}), \cdot) = \frac{1}{\sqrt{2}}\big(k(\cdot, \mathbf{x}^{(\ell)}) \otimes k(\cdot, \mathbf{x}^{(r)}) - k(\cdot, \mathbf{x}^{(r)}) \otimes k(\cdot, \mathbf{x}^{(\ell)})\big) \tag{9}$$

where $\otimes$ are the usual tensor product. Recall we defined the preferential value function for items as,

$$\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}(g) = \mathbb{E}[g(X^{(\ell)}, X^{(r)}) \mid X^{(\ell)}_S = \mathbf{x}^{(\ell)}_S, X^{(r)}_S = \mathbf{x}^{(r)}_S] \tag{10}$$

as $\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}$ is a bounded linear functional on g where $g \in \mathcal{H}_{k_E}$ is bounded, Riesz representation theorem [25] tells us there exists a Riesz representation of the functional in $\mathcal{H}_{k_E}$, which for notation simplicity, we will denote it as $\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}$ as well. This corresponds to,

$$\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}(g) = \mathbb{E}[g(X^{(\ell)}, X^{(r)}) \mid X^{(\ell)}_S = \mathbf{x}^{(\ell)}_S, X^{(r)}_S = \mathbf{x}^{(r)}_S] \tag{11}$$

$$= \langle g, \mathbb{E}[k_E\left((X^{(\ell)}, X^{(r)}), \cdot\right) \mid X^{(\ell)}_S = \mathbf{x}^{(\ell)}_S, X^{(r)}_S = \mathbf{x}^{(r)}_S]\rangle_{\mathcal{H}_{k_E}} \tag{12}$$

$$= \langle g, \nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}\rangle_{\mathcal{H}_{k_E}} \tag{13}$$

now we expand the expectation of the feature map as,

$$\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S} = \mathbb{E}\left[\frac{1}{\sqrt{2}}\big(k(\cdot,X^{(\ell)})\otimes k(\cdot,X^{(r)}) - k(\cdot,X^{(r)})\otimes k(\cdot,X^{(\ell)}) \mid X^{(\ell)}_S = \mathbf{x}^{(\ell)}_S, X^{(r)}_S = \mathbf{x}^{(r)}_S\right] \tag{14}$$

However, we note that

$$\mathbb{E}[k(\cdot,X^{(\ell)})\otimes k(\cdot,X^{(r)}) \mid X^{(\ell)}_S = \mathbf{x}^{(\ell)}_S, X^{(r)}_S = \mathbf{x}^{(r)}_S] = \mathbb{E}[k(\cdot,X) \mid X_S = \mathbf{x}^{(\ell)}_S] \otimes \mathbb{E}[k(\cdot,X) \mid X_S = \mathbf{x}^{(r)}_S],$$

because $X^{(\ell)}$ and $X^{(r)}$ are identical copies of X and we take the reference distribution as $p(X^{(\ell)},X^{(r)} \mid X^{(\ell)}_S = \mathbf{x}^{(\ell)}_S, X^{(r)}_S = \mathbf{x}^{(r)}_S) = p(X^{(\ell)} \mid X^{(\ell)}_S = \mathbf{x}^{(\ell)}_S)p(X^{(r)} \mid X^{(r)}_S = \mathbf{x}^{(r)}_S)$. Focusing on the duplicating component, we have,

$$\begin{aligned}
\mathbb{E}[k(\cdot,X) \mid X_S = \mathbf{x}^{(\ell)}] &= \mathbb{E}[k_S(\cdot,X_S)\otimes k_{S^c}(\cdot,X_{S^c}) \mid X_S = \mathbf{x}^{(\ell)}_S] && (15)\\
&= k_S(\cdot,\mathbf{x}^{(\ell)}_S)\otimes \mathbb{E}[k_{S^c}(\cdot,X_{S^c}) \mid X_S = \mathbf{x}^{(\ell)}_S] && (16)\\
&= k_S(\cdot,\mathbf{x}^{(\ell)}_S)\otimes \mu_{X_{S^c}|X_S=\mathbf{x}^{(\ell)}_S} && (17)\\
&=: \mathcal{K}(\mathbf{x}^{(\ell)},S) && (18)
\end{aligned}$$

therefore by symmetry, we can arrange the terms in Eq 14 and conclude the proposition,

$$\nu^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S} = \frac{1}{\sqrt{2}}\Big(\mathcal{K}(\mathbf{x}^{(\ell)},S)\otimes\mathcal{K}(\mathbf{x}^{(r)},S) - \mathcal{K}(\mathbf{x}^{(r)},S)\otimes\mathcal{K}(\mathbf{x}^{(\ell)},S)\Big) \tag{19}$$

□

To estimate the preferential value functional, we simply replace the conditional mean embeddings with the empirical versions, i.e. $\hat{\mathcal{K}}(\mathbf{x},S) = k_S(\cdot,\mathbf{x}_S)\otimes\hat{\mu}_{X_{S^c}|X_S=\mathbf{x}_S}$, where $\hat{\mu}_{X_{S^c}|X_S=\mathbf{x}_S} = \mathbf{K}_{\mathbf{x}_S,\mathbf{x}_S}(\mathbf{K}_{\mathbf{x}_S,\mathbf{x}_S}+n\lambda I)^{-1}\mathbf{\Phi}^\top_{X_{S^c}}$ is the standard conditional mean embedding estimator ($\mathbf{\Phi}_{X_{S^c}}$ is the feature map matrix of rv X_{S^c}).

Now we proceed to estimate the preferential value function given a function g from the RKHS,

Proposition 3.2 (Non-parametric Estimation)**.** *Given $\hat{g} = \sum_{j=1}^m \alpha_j k_E((\mathbf{x}^{(\ell)}_j,\mathbf{x}^{(r)}_j),\cdot)$, datasets $\mathbf{X}^{(\ell)},\mathbf{X}^{(r)}$, test items $\mathbf{x}^{(\ell)},\mathbf{x}^{(r)}$, the preferential value function at test items $\mathbf{x}^{(\ell)},\mathbf{x}^{(r)}$ for coalition S and preference function $\hat{g}$ can be estimated as*

$$\hat{\nu}^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}(\hat{g}) = \boldsymbol{\alpha}^\top\left(\Gamma(\mathbf{X}^{(\ell)}_S,\mathbf{x}^{(\ell)}_S)\odot\Gamma(\mathbf{X}^{(r)}_S,\mathbf{x}^{(r)}_S) - \Gamma(\mathbf{X}^{(\ell)}_S,\mathbf{x}^{(r)}_S)\odot\Gamma(\mathbf{X}^{(r)}_S,\mathbf{x}^{(\ell)}_S)\right),$$

where $\Gamma(\mathbf{X}^{(\ell)}_S,\mathbf{x}^{(\ell)}_S) = \mathbf{K}_{\mathbf{X}^{(\ell)}_S,\mathbf{x}^{(\ell)}_S}\odot\mathbf{K}_{\mathbf{X}^{(\ell)}_{S^c},\mathbf{X}_{S^c}}\mathbf{K}^{-1}_{\mathbf{X}_S,\lambda}\mathbf{K}_{\mathbf{X}_S,\mathbf{x}^{(\ell)}_S}, \mathbf{K}_{\mathbf{X}_S,\lambda} = \mathbf{K}_{\mathbf{X}_S,\mathbf{X}_S}+n\lambda I$, $\boldsymbol{\alpha} = \{\alpha_j\}_{j=1}^m$ and $\lambda > 0$ is a regularisation parameter.

Proof. Given $\hat{g}$, the preferential value function evaluated at $\hat{g}$ can be written as,

$$\begin{aligned}
\hat{\nu}^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}(\hat{g}) &= \langle\hat{g},\hat{\nu}^{(p_I)}_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S}\rangle_{\mathcal{H}_{k_E}} && (20)\\
&= \langle\sum_{j=1}^m \alpha_j k_E((\mathbf{x}^{(\ell)}_j,\mathbf{x}^{(r)}_j),\cdot), \frac{1}{\sqrt{2}}\left(\hat{\mathcal{K}}(\mathbf{x}^{(\ell)},S)\otimes\hat{\mathcal{K}}(\mathbf{x}^{(r)},S) - \hat{\mathcal{K}}(\mathbf{x}^{(r)},S)\otimes\hat{\mathcal{K}}(\mathbf{x}^{(\ell)},S)\right)\rangle_{\mathcal{H}_{k_E}} && (21)\\
&= \frac{1}{\sqrt{2}}\left\langle\sum_{j=1}^m \alpha_j k_E((\mathbf{x}^{(\ell)}_j,\mathbf{x}^{(r)}_j),\cdot), \hat{\mathcal{K}}(\mathbf{x}^{(\ell)},S)\otimes\hat{\mathcal{K}}(\mathbf{x}^{(r)},S)\right\rangle && (22)\\
&\quad - \frac{1}{\sqrt{2}}\left\langle\sum_{j=1}^m \alpha_j k_E((\mathbf{x}^{(\ell)}_j,\mathbf{x}^{(r)}_j),\cdot), \hat{\mathcal{K}}(\mathbf{x}^{(r)},S)\otimes\hat{\mathcal{K}}(\mathbf{x}^{(\ell)},S)\right\rangle && (23)
\end{aligned}$$

Now we focus on the first component, and rewrite:

$$\frac{1}{\sqrt{2}}\left\langle \sum_{j=1}^{m} \alpha_j k_E((\mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot), \hat{\mathcal{K}}(\mathbf{x}^{(\ell)}, S) \otimes \hat{\mathcal{K}}(\mathbf{x}^{(r)}, S) \right\rangle = \sum_{j=1}^{m} A_j^{(1)} \tag{24}$$

and we continue to expand the terms,

$$A_j^{(1)} := \frac{1}{\sqrt{2}}\left\langle \alpha_j k_E((\mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot), \hat{\mathcal{K}}(\mathbf{x}^{(\ell)}, S) \otimes \hat{\mathcal{K}}(\mathbf{x}^{(r)}, S) \right\rangle \tag{25}$$

$$= \frac{1}{\sqrt{2}}\left\langle \frac{\alpha_j}{\sqrt{2}}\big(k(\cdot, \mathbf{x}_j^{(\ell)}) \otimes k(\cdot, \mathbf{x}_j^{(r)}) - k(\cdot, \mathbf{x}_j^{(r)}) \otimes k(\cdot, \mathbf{x}_j^{(\ell)})\big), \hat{\mathcal{K}}(\mathbf{x}^{(\ell)}, S) \otimes \hat{\mathcal{K}}(\mathbf{x}^{(r)}, S) \right\rangle \tag{26}$$

$$= \frac{\alpha_j}{2}\left(\left\langle k(\cdot, \mathbf{x}_j^{(\ell)}) \otimes k(\cdot, \mathbf{x}_j^{(r)}), \hat{\mathcal{K}}(\mathbf{x}^{(\ell)}, S) \otimes \hat{\mathcal{K}}(\mathbf{x}^{(r)}, S) \right\rangle - \left\langle k(\cdot, \mathbf{x}_j^{(r)}) \otimes k(\cdot, \mathbf{x}_j^{(\ell)}), \hat{\mathcal{K}}(\mathbf{x}^{(\ell)}, S) \otimes \hat{\mathcal{K}}(\mathbf{x}^{(r)}, S) \right\rangle\right) \tag{27}$$

$$= \frac{\alpha_j}{2}\left(A_j^{(1,\ell)} - A_j^{(1,r)}\right) \tag{28}$$

We then note that

$$A_j^{(1,\ell)} := \left\langle k(\cdot, \mathbf{x}_j^{(\ell)})) \otimes k(\cdot, \mathbf{x}_j^{(r)}), \hat{\mathcal{K}}(\mathbf{x}^{(\ell)}, S) \otimes \hat{\mathcal{K}}(\mathbf{x}^{(r)}, S) \right\rangle \tag{29}$$

$$= \left\langle k(\cdot, \mathbf{x}_j^{(\ell)}), \hat{\mathcal{K}}(\mathbf{x}^{(\ell)}, S) \right\rangle \left\langle k(\cdot, \mathbf{x}_j^{(r)}), \hat{\mathcal{K}}(\mathbf{x}^{(r)}, S) \right\rangle \tag{30}$$

$$= k_S(\mathbf{x}_j^{(\ell)}, \mathbf{x}^{(\ell)}) \mathbf{K}_{\mathbf{x}_j^{(\ell)}{}_S, \mathbf{X}_S} (\mathbf{K}_{\mathbf{X}_S, \mathbf{X}_S} + n\lambda I)^{-1} \mathbf{K}_{\mathbf{X}_{S^c}, \mathbf{x}_{S^c}^{(\ell)}} \tag{31}$$

$$\times k_S(\mathbf{x}_j^{(r)}, \mathbf{x}^{(r)}) \mathbf{K}_{\mathbf{x}_j^{(r)}{}_S, \mathbf{X}_S} (\mathbf{K}_{\mathbf{X}_S, \mathbf{X}_S} + n\lambda I)^{-1} \mathbf{K}_{\mathbf{X}_{S^c}, \mathbf{x}_{S^c}^{(r)}} \tag{32}$$

$$= \Gamma(\mathbf{x}_{jS}^{(\ell)}, \mathbf{x}_S^{(\ell)}) \odot \Gamma(\mathbf{x}_{jS}^{(r)}, \mathbf{x}_S^{(r)}) \tag{33}$$

To go from the second equation to the third equation in this paragraph, realise $k(\cdot, \mathbf{x}^{(\ell)}) = k_S(\cdot, \mathbf{x}_S^{(\ell)}) \otimes k_{S^c}(\cdot, \mathbf{x}_{S^c}^{(\ell)})$ by product kernel assumption. In this case, we can rewrite $A_j^{(1)}$ as,

$$A_j^{(1)} = \frac{\alpha_j}{2}\left(\Gamma(\mathbf{x}_{jS}^{(\ell)}, \mathbf{x}_S^{(\ell)}) \odot \Gamma(\mathbf{x}_{jS}^{(r)}, \mathbf{x}_S^{(r)}) - \Gamma(\mathbf{x}_{jS}^{(r)}, \mathbf{x}_S^{(\ell)}) \odot \Gamma(\mathbf{x}_{jS}^{(\ell)}, \mathbf{x}_S^{(r)})\right) \tag{34}$$

Analogously, define $\sum A_j^{(2)}$ as the second component after the subtraction sign, by symmetry, we know

$$A_j^{(2)} = \frac{\alpha_j}{2}\left(\Gamma(\mathbf{x}_{jS}^{(\ell)}, \mathbf{x}_S^{(r)}) \odot \Gamma(\mathbf{x}_{jS}^{(r)}, \mathbf{x}_S^{(\ell)}) - \Gamma(\mathbf{x}_{jS}^{(r)}, \mathbf{x}_S^{(r)}) \odot \Gamma(\mathbf{x}_{jS}^{(\ell)}, \mathbf{x}_S^{(\ell)})\right) \tag{35}$$

by subtracting $A_j^{(1)}$ and $A_j^{(1)}$, we get the following:

$$\hat{\nu}^{(p_I)}_{\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}, S}(\hat{g}) = \sum_{j=1}^{m} \alpha_j \left(\Gamma(\mathbf{x}_{jS}^{(\ell)}, \mathbf{x}_S^{(\ell)}) \odot \Gamma(\mathbf{x}_{jS}^{(r)}, \mathbf{x}_S^{(r)}) - \Gamma(\mathbf{x}_{jS}^{(r)}, \mathbf{x}_S^{(\ell)}) \odot \Gamma(\mathbf{x}_{jS}^{(\ell)}, \mathbf{x}_S^{(r)})\right) \tag{36}$$

writing it in compact form, we arrive to our result,

$$= \boldsymbol{\alpha}^\top \left(\Gamma(\mathbf{X}_S^{(\ell)}, \mathbf{x}_S^{(\ell)}) \odot \Gamma(\mathbf{X}_S^{(r)}, \mathbf{x}_S^{(r)}) - \Gamma(\mathbf{X}_S^{(\ell)}, \mathbf{x}_S^{(r)}) \odot \Gamma(\mathbf{X}_S^{(r)}, \mathbf{x}_S^{(\ell)})\right) \tag{37}$$

□

Proposition 3.3 (Preferential value function for contexts)**.** *Given a preference function $g_U \in \mathcal{H}_{k_E^U}$, denote $\Omega' = \{1, ..., d'\}$, then the utility of context features $S' \subseteq \Omega'$ on $\{\mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\}$ is measured by $\nu^{(p_U)}_{\mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}, S'}(g_U) = \mathbb{E}[g_U\left(\{\mathbf{u}_{S'}, U_{S'^c}\}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right) \mid U_{S'} = \mathbf{u}_{S'}]$ where the expectation is taken over the observational distribution of U. Now, given a test triplet $(\mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)})$, if $\hat{g}_U = \sum_{j=1}^{m} \alpha_j k_E^U\left((\mathbf{u}_j, \mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot\right)$, the non-parametric estimator is:*

$$\hat{\nu}^{(p_U)}_{\mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}, S'}(\hat{g}_U) = \boldsymbol{\alpha}^\top \left(\left(\mathbf{K}_{\mathbf{U}_{S'}, \mathbf{u}_{S'}} \odot \mathbf{K}_{\mathbf{U}_{S'^c}, \mathbf{U}_{S'^c}} \left(\mathbf{K}_{\mathbf{U}_{S'}, \mathbf{U}_{S'}} + m\lambda' I\right)^{-1} \mathbf{K}_{\mathbf{U}_{S'}, \mathbf{u}_{S'}}\right) \odot \Xi_{\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}}\right)$$

where $\Xi_{\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}} = \left(\mathbf{K}_{\mathbf{X}^{(\ell)}, \mathbf{x}^{(\ell)}} \odot \mathbf{K}_{\mathbf{X}^{(r)}, \mathbf{x}^{(r)}} - \mathbf{K}_{\mathbf{X}^{(r)}, \mathbf{x}^{(\ell)}} \odot \mathbf{K}_{\mathbf{X}^{(\ell)}, \mathbf{x}^{(r)}}\right)$.

Proof. Recall the feature map of the kernel k_E^U takes the following form,

$$k_E^U\left((\mathbf{u}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}), \cdot\right) = k_u(\mathbf{u}, \cdot) \otimes k_E((\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}), \cdot) \tag{38}$$

Therefore we can express the preferential value function for context as,

$$\nu^{(p_U)}_{\mathbf{u},\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S'}(g_U) = \mathbb{E}\left[g_U\left(\{\mathbf{u}_{S'}, U_{S^c}\}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}\right) \mid U_S = \mathbf{u}_{S'}\right] \tag{39}$$

$$= \left\langle g_U, \mathbb{E}\left[k_E^U\left((\{\mathbf{u}_{S'}, U_{S'^c}\}, \mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}), \cdot\right) \mid U_{S'} = \mathbf{u}_{S'}\right]\right\rangle \tag{40}$$

$$= \left\langle g_U, \mathbb{E}\left[k_u(\{\mathbf{u}_{S'}, U_{S^c}\} \mid U_{S'} = \mathbf{u}_{S'}\right] \otimes k_E((\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}), \cdot)\right\rangle \tag{41}$$

$$= \left\langle g_U, k_{u_{S'}}(\mathbf{u}_{S'}) \otimes \mu_{U_{S'^c}|U_{S'}=\mathbf{u}_{S'}} \otimes k_E((\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}), \cdot)\right\rangle \tag{42}$$

The remaining steps are analogous to [12, Prop.2]. To obtain the empirical estimation, we first replace the conditional mean embedding $\mu_{U_{S'^c}|U_{S'}=\mathbf{u}_{S'}}$ with its empirical estimate and replace g_U with $\hat{g}_U = \sum_{j=1}^m \alpha_j k_E^U\left((\mathbf{u}_j, \mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot\right)$. Now the empirical estimator has the following form,

$$\hat{\nu}^{(p_U)}_{\mathbf{u},\mathbf{x}^{(\ell)},\mathbf{x}^{(r)},S'}(\hat{g}_U) = \left\langle \sum_{j=1}^m \alpha_j k_E^U\left((\mathbf{u}_j, \mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot\right), k_{u_{S'}}(\mathbf{u}_{S'}) \otimes \hat{\mu}_{U_{S'^c}|U_{S'}=\mathbf{u}_{S'}} \otimes k_E((\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}), \cdot)\right\rangle \tag{43}$$

$$= \sum_{j=1}^m \alpha_j \left\langle k_u(\mathbf{u}_j, \cdot) \otimes k_E((\mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), \cdot), k_{u_{S'}}(\mathbf{u}_{S'}) \otimes \hat{\mu}_{U_{S'^c}|U_{S'}=\mathbf{u}_{S'}} \otimes k_E((\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)}), \cdot)\right\rangle \tag{44}$$

$$= \sum_{j=1}^m \alpha_j \left\langle k_u(\mathbf{u}_j, \cdot), k_{u_S}(\mathbf{u}_S) \otimes \hat{\mu}_{U_{S'^c}|U_S=\mathbf{u}_S}\right\rangle k_E\left((\mathbf{x}_j^{(\ell)}, \mathbf{x}_j^{(r)}), (\mathbf{x}^{(\ell)}, \mathbf{x}^{(r)})\right) \tag{45}$$

Now write everything in terms of matrices,

$$= \boldsymbol{\alpha}^\top \left(\left(\mathbf{K}_{\mathbf{U}_{S'},\mathbf{u}_{S'}} \odot \mathbf{K}_{\mathbf{U}_{S'^c},\mathbf{U}_{S'^c}} \left(\mathbf{K}_{\mathbf{U}_{S'},\mathbf{U}_{S'}} + m\lambda' I\right)^{-1} \mathbf{K}_{\mathbf{U}_{S'},\mathbf{u}_{S'}}\right) \odot \Xi_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)}}\right) \tag{46}$$

where $\Xi_{\mathbf{x}^{(\ell)},\mathbf{x}^{(r)}} = \left(\mathbf{K}_{\mathbf{X}^{(\ell)},\mathbf{x}^{(\ell)}} \odot \mathbf{K}_{\mathbf{X}^{(r)},\mathbf{x}^{(r)}} - \mathbf{K}_{\mathbf{X}^{(r)},\mathbf{x}^{(\ell)}} \odot \mathbf{K}_{\mathbf{X}^{(\ell)},\mathbf{x}^{(r)}}\right)$. □