Who to imitate: Imitating desired behavior from diverse multi-agent datasets

Tim Franzmeyer [1] **Jakob N. Foerster** [1] **Edith Elkind** [1] **Philip H.S. Torr** [1] **João F. Henriques** [1]

Abstract

AI agents are commonly trained with large datasets of unfiltered demonstrations of human behavior. However, not all behaviors are equally safe or desirable. We assume that desired traits for an AI agent can be approximated by a *desired value function* (DVF), that assigns scores to collective outcomes in the dataset. For example, in a dataset of vehicle interactions, the DVF might refer to the number of occurred incidents. We propose to first assess how well individual agents' behavior is aligned with the DVF, e.g., assessing how likely an agent is to cause incidents, to then only imitate agents with desired behaviors. To identify agents with desired behavior, we propose the concept of an agent's *Exchange Value*, which quantifies the expected change in collective value when substituting the agent into a random group. This concept is similar to Shapley Values used in Economics, but offers greater flexibility. We further introduce a variance maximization objective to compute Exchange Values from incomplete observations, effectively clustering agents by their unobserved traits. Using both human and simulated datasets, we learn aligned imitation policies that outperform relevant baselines.

1. Introduction

Learning imitation policies for AI agents from large datasets of human behavior is a promising approach for successful human-AI and AI-AI interaction, even in complex cooperative environments [4, 8, 13, 31]. However, unfiltered datasets of human interactions often contain behaviors that may be undesirable for AI agents. This work assumes that it is possible to approximately score how desirable behavior is through a (given) *desired value function* (DVF)[1]. We assume that the DVF assigns a score to each trajectory in the dataset (unlike a per-state reward function), which is, e.g., the case when desirability cannot practically be assessed at a per-state level, or when costly human annotations are only obtained for overall outcomes [33]. Such DVFs have been successfully applied in fine-tuning of large language models [1, 26, 33].

This work considers imitating a multi-agent dataset while ensuring that the learned policy is aligned with a DVF that is only defined for groups of agents, i.e. for collective behavior. Such collective value functions assign scores to multi-agent trajectories and are relevant in real-world scenarios where outcomes depend on complex interactions of multiple agents. Consider, for example, the scenario of imitating a dataset of vehicle interactions while ensuring not to imitate behavior that is likely to cause incidents. Another example constitutes imitating a dataset of a multi-agent online game, where an AI designer wants to ensure that no behavior is imitated which decreases players' satisfaction. In both scenarios, the DVF is only defined for collective outcomes – assigning either the number of occurred incidents or the average satisfaction of players to group trajectories in the datasets.

Understanding whether the behavior of individual agents in a dataset is aligned with a DVF requires assessing individual contributions to the DVF. This credit assignment problem [30] is challenging in real-world datasets for three reasons. First, groups of certain sizes may never be observed, as many environments only permit specific group sizes. This makes Shapley Values [30] – a concept commonly used in Economics for credit assignment – inapplicable here as it relies on the comparisons of groups of different sizes. Second, real-world datasets for large groups are necessarily incomplete, i.e. do not contain observations for all (combinatorially many) possible groups of agents. Third, datasets of human interactions may be *fully anonymized* by assigning single-use agent IDs. In such fully-anonymized datasets, each agent is observed only as part of one multi-agent trajectory, hence only appearing once in the entire dataset. In this case, the credit assignment problem is degenerate, and requires incorporating information about agents' low-level behavior.

Inspired by Shapley Values, we propose the concept of Exchange Values (EVs) of individual agents to address these

[1]University of Oxford, UK. Correspondence to: Tim Franzmeyer <frtim at robots dot ox dot ac dot uk>.

[1]The DVF itself is not sufficient to describe desired behavior completely, as it possibly only covers a subset of behavior, e.g., safety-relevant aspects. It is complementary to the more complex and nuanced behaviors that are obtained by imitating human demonstrations, providing guardrails or additional guidance.

issues. EVs define the contribution of a given agent to a collective value function as the expected *change in value* when exchanging an agent in a random group for the given agent. Hence, EVs only require comparing groups of equal sizes and can be applied to datasets where some group sizes never occur. We formally characterize the relationship between EVs and Shapley Values. To estimate EVs of individual agents also from incomplete datasets, we propose a novel method that assigns agents to clusters such that the variance in EVs is maximized. Under the simplifying assumption of an additive collective value function, we theoretically show that this variance maximization objective is equivalent to clustering agents by their unobserved traits. Lastly, we modify the variance maximization objective to also account for agents' low-level behavior, which allows us to estimate EVs from fully-anonymized datasets.

Using both simulated datasets and datasets of human interactions, we empirically demonstrate that EVs allow to assess the contributions of individual agents to a given DVF, including successful application to a fully-anonymized human dataset. We then propose to imitate large unfiltered datasets by only imitating the behavior of agents with a positive contribution to the DVF. This approach allows learning from interactions with agents with undesired behaviors (without imitating them), in contrast to simply excluding all trajectories with a low collective score from the training dataset. We find that our method enables imitating diverse datasets while ensuring alignment with relevant DVFs. Our approach outperforms relevant baselines, such as excluding all trajectories with a low collective score or framing the problem as offline reinforcement learning.

Our work makes the following contributions:

- We introduce Exchange Values to compute an agent's individual contribution to a collective value function and show their relation to Shapley Values.
- We propose a novel variance maximization method to estimate contributions from incomplete datasets and show a theoretical connection to clustering by unobserved traits.
- We empirically demonstrate how EVs can be estimated from fully-anonymized data and utilized to learn policies aligned with a DVF using imitation learning.

2. Related Work

Most previous work on aligning AI agents' policies with desired value functions either relies on simple hand-crafted rules [39, 8], which do not scale to complex environments, or performs post-processing of imitation policies with fine-tuning [33, 26, 10, 1], which requires access to the environment or a simulator. In language modeling, Korbak et al. [18] showed that accounting for the alignment of behavior with the DVF already during imitation learning yields results superior to fine-tuning after-the-fact, however their approach considers an agent-specific value function. In contrast, we consider learning a policy aligned with a collective value function, and from offline data alone.

Credit assignment in multi-agent systems was initially studied in Economics [30]. Subsequently, Shapley Values [30] and related concepts have been applied in multi-agent reinforcement learning, to distribute rewards among individual agents during the learning process [5, 9, 24, 38, 21, 37]. Outside of policy learning, Heuillet et al. [14] used Shapley values to analyze agent contributions in multi-agent environments, however this requires privileged access to a simulator, in order to replace agents with randomly-acting agents. In contrast to Shapley Values, the proposed Exchange Values can be applied in the case of infeasible coalition sizes, omitting, e.g., simulating infeasible coalitions by summing over multiple outcomes or with random-action policies.

In contrast to this work, existing research on multi-agent imitation learning typically assumes observations to be generated by optimal agents, as well as simulator access [20, 32, 41]. In a related setting, offline multi-agent reinforcement learning [16, 36, 35] shares similarities with our problem, involving policy learning from multi-agent demonstrations using offline data alone. In contrast to our setting, offline multi-agent reinforcement learning assumes a dense reward signal to be given, while the DVF assigns a single score per trajectory.

In single-agent settings, a large body of work investigates estimating demonstrator expertise for imitation learning [6, 42, 3, 29, 2, 40], which however does not translate to multi-agent settings due to the challenge of credit assignment.

To the best of our knowledge, no prior work has considered imitating multi-agent datasets generated by agents with varying behaviors while ensuring alignment with a collective value function.

3. Background and Notation

Markov Game. We consider Markov Games [22], which generalize Markov Decision Processes (MDP) to multi-agent scenarios. In a Markov Game, agents interact in a common environment. At time step t, each agent (the ith of a total of m agents) takes the action a_i^t and the environment transitions from state s^t to s^{t+1}. A reduced Markov game (without rewards) is then defined by a state space $\mathcal{S}$ ($s^t \in \mathcal{S}$), a distribution of initial states η, the action space $\mathcal{A}_i$ ($a_i^t \in \mathcal{A}_i$) of each agent i, an environment state transition probability $P(s^{t+1}|s^t, a_1, \ldots, a_m)$ and the episode length T. We denote this Markov Game as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, T)$. We define a trajectory as $\tau = (s_0, \mathbf{a}_0, s_1, \mathbf{a}_1, \ldots, s_T)$.

Set of multi-agent demonstrations generated by many agents. We consider a symmetric Markov Game $\mathcal{M}$ involving m agents, where $N = 1, \ldots, n$ is the set of all demonstrator agents and $n \geq m$. The dataset D is constructed from a variety of coalitions interacting in $\mathcal{M}$. These coalitions, or groups of agents, are referred to interchangeably. The set of observed coalitions is denoted as Q, comprising q coalitions C where $C \in Q$. Each coalition C is a subset of N and consists of m agents. The dataset $\mathcal{D}$ comprises tuples $(\bar{C}, \tau)$, where $\bar{C}$ represents an observed coalition from Q, and τ is the trajectory generated by that coalition in $\mathcal{M}$. In the context of the Markov Game, each state s_t in the tractory τ describes the collective state of all individual agents in the environment.

Shapley Values. We now define the concept of the Shapley Value of an agent [30], which is commonly used to evaluate contributions of individual agents to a collective value function in a characteristic function game. Definition 3.2 below is somewhat non-conventional, but can be easily seen to be equivalent to the standard definition.

Definition 3.1 (Characteristic function game)**.** A characteristic function game G is given by a pair (N, v), where $N = \{1, \ldots, n\}$ is a finite, non-empty set of agents and $v : 2^N \to \mathbb{R}$ is a characteristic function, which maps each coalition $C \subseteq N$ to a real number $v(C)$; it is assumed that $v(\emptyset) = 0$. The number $v(C)$ is usually referred to as the value of the coalition C.

Given a characteristic function game $G = (N, v)$, let $\Pi_{N\setminus\{i\}}$ denote the set of all permutations of $N\setminus\{i\}$, i.e., one-to-one mappings from $N\setminus\{i\}$ to itself. For each permutation $\pi \in \Pi_{N\setminus\{i\}}$, we denote by $S_\pi(m)$ the slice of π up until and including position m; we think of $S_\pi(m)$ as the set of all agents that appear in the first m positions in π (note that $S_\pi(0) = \emptyset$). The marginal contribution of an agent i with respect to a permutation π and a slice m in a game $G = (N, v)$ is given by

$$\Delta^G_{m,\pi}(i) = v(S_\pi(m) \cup \{i\}) - v(S_\pi(m)).$$

This quantity measures the increase in the value of the coalition consisting of the agents in slice m of π when agent i joins them. We can now define the Shapley Value of an agent i: it is simply the agent's average marginal contribution, where the average is taken over all permutations of $N\setminus\{i\}$ and all slices.

Definition 3.2 (Shapley Value)**.** Given a characteristic function game $G = (N, v)$ with $|N| = n$, the Shapley Value of an agent $i \in N$ is denoted by $\varphi_i(G)$ and is given by

$$\varphi_i(G) = {}^1\!/_{n!} \cdot \sum_{m=0}^{n-1} \sum_{\pi \in \Pi_{N\setminus\{i\}}} \Delta^G_{m,\pi}(i). \qquad (1)$$

Def. 3.2 is important in the context of credit assignment, as it provides a possible solution for distributing collective value to individual agents. It also has several properties related to its consistency [30].

4. Methods

4.1. Exchange Values to evaluate agents' individual contributions

Overview and notation. When agents form coalitions, Shapley Values are often used to evaluate all agents' individual contributions. Now, as per Definition 3.2, the Shapley Value of agent i is determined by the change in value when *adding* agent i to a given coalition. This requires evaluating the values of coalitions of different sizes, as adding an agent increases the coalition size by one. However, many relevant real-world coalition formation scenarios only permit specific coalition sizes, potentially even only a single size. This is the case, for example, for football games, where a team (coalition) always has 11 players, hence Shapley values cannot be computed, as the game outcome cannot be evaluated for 10 or 12 players. To evaluate agents' contributions in games that only permit certain coalition sizes, we first define the concept of Exchange Values for regular characteristic function games. We then show that our definition extends naturally to characteristic function games with constraints on feasible coalition sizes.

In words, the exchange contribution of an agent i with respect to a permutation and slice is defined as the change in value when replacing the last agent in the slice of a permutation by agent i. Specifically, we define the exchange contribution $\Gamma^G_{m,\pi}(i)$ of an agent i with respect to a permutation π and slice m in a game $G = (N, v)$ as

$$\Gamma^G_{m,\pi}(i) = v(S_\pi(m-1) \cup \{i\}) - v(S_\pi(m)).$$

Note that this quantity is computed as a difference between values of equal-size coalitions, unlike the marginal contribution $\Delta^G_{m,\pi}(i)$ in the definition of Shapley Values.

We can now define the Exchange Value analogously to the Shapley Value as the average exchange contribution over all permutations of $N\setminus\{i\}$ and all non-empty slices.

Definition 4.1 (Exchange Value)**.** Given a characteristic function game $G = (N, v)$ with $|N| = n$, the Exchange Value of an agent $i \in N$ is denoted by $\gamma_i(G)$ and is given by

$$\gamma_i(G) = ((n-1)! \cdot (n-1))^{-1} \cdot \sum_{m=1}^{n-1} \sum_{\pi \in \Pi_{N\setminus\{i\}}} \Gamma^G_{m,\pi}(i). \qquad (2)$$

Relation between Shapley Value and Exchange Value. By rearranging the terms, it can be verified that the Exchange Value of an agent can be derived from its Shapley Value by subtracting a fraction of the value of the grand

coalition N, i.e.,

$$\gamma_i(G) = \phi_i(G) - 1/n \cdot v(N).$$

Intuitively, this is because (1) no exchange comparisons can be made for the grand coalition, as all agents are already part of the grand coalition; (2) every non-empty coalition appears the same number of times (and with the same sign) in summations (1) and (2). Consequently, the *ordering* of the agents is equivalent under Shapley Values and Exchange Values, as the value of the grand coalition is independent of i. Moreover, since $\sum_{i \in N} \phi_i(G) = v(N)$, we have $\sum_{i \in N} \gamma_i(G) = \sum_{i \in N} \left(\phi_i(G) - \frac{1}{n} \cdot v(N)\right) = 0$. In words, the sum of all agents' Exchange Values is zero. It can also be shown that Exchange Values satisfy the symmetry axiom: if $v(S \cup \{i\}) = v(S \cup \{j\})$ for all $S \subseteq N \setminus \{i, j\}$, we have $\gamma_i(G) = \gamma_j(G)$.

4.2. Computing Exchange Values if only certain coalition sizes are feasible

For a characteristic function game $\mathcal{G} = (N, v)$ the value function v can be evaluated for each possible coalition $C \subseteq N$. We now consider the case where the value function v is only defined for coalitions of certain sizes $m \in M$, i.e. v is only defined for a subset of coalitions of certain sizes.

Definition 4.2 (Constrained characteristic function game)**.** A constrained characteristic function game $\bar{G}$ is given by a tuple (N, v, M), where $N = \{1, \dots, n\}$ is a finite, non-empty set of agents, $M \subseteq \{0, \dots, n-1\}$ is a set of feasible coalition sizes and $v : \{C \in 2^N : |C| \in M\} \to \mathbb{R}$ is a characteristic function, which maps each coalition $C \subseteq N$ of size $|C| \in M$ to a real number $v(C)$.

Note that both the Shapley Value and the Exchange Value are generally undefined for constrained characteristic function games, as the value function is not defined for coalitions C of size $|C| \notin M$. The definition of the Shapley Value cannot easily be adapted to constrained characteristic function games, as its computation requires evaluating values of coalitions of different sizes. This becomes clear when considering the case of a constrained characteristic function game that only permits a single coalition size (see Definition 3.2). In contrast, the definition of the Exchange Value can be straightforwardly adapted to constrained characteristic function games by limiting the summation to slices of size $m \in M^+$, where $M^+ = \{m \in M : m > 0\}$. Hence, we define the Constrained Exchange Value as the average exchange contribution over all permutations of $N \backslash \{i\}$ and over all slices of size $m \in M^+$.

Definition 4.3 (Constrained Exchange Value)**.** Given a constrained characteristic function game $\bar{G} = (N, v, M)$ with $|N| = n$, the Constrained Exchange Value of an agent $i \in N$ is denoted by $\gamma_i(\bar{G})$ and is given by

$$\gamma_i(\bar{G}) = ((n-1)! \cdot |M^+|)^{-1} \cdot \sum_{m \in M^+} \sum_{\pi \in \Pi_{N \setminus \{i\}}} \Gamma^{\bar{G}}_{m,\pi}(i).$$

Note that the Constrained Exchange Value is equivalent to the Exchange Value if $M = \{1, \dots, n-1\}$.

4.3. Estimating Exchange Values by sampling coalitions

We can achieve an unbiased estimate of the Constrained Exchange Value by sampling coalition sizes m uniformly at random from M and sampling permutations π uniformly at random from $\Pi_{N \setminus \{i\}}$, as the Constrained Exchange Value can also be defined as

$$\gamma_i(\bar{G}) = \mathbb{E}_{m \sim U(M^+), \pi \sim U(\Pi_{N \setminus \{i\}})} \left[\Gamma^{\bar{G}}_{m,\pi}(i)\right].$$

In the limit of infinite samples, the expectation converges to the true Constrained Exchange Value. This is relevant for real datasets, where one may not have samples from all possible coalitions, but rather a uniformly-sampled subset.

4.4. Estimating Exchange Values with clustering

We now consider the case where we can expect multiple agents in N to be behaviorally similar, hence having similar Exchange Value. To better estimate Exchange Values in the case of incomplete observations, we propose to assign agents to clusters $K = \{1, \dots, k-1\}$, and then assign equal Exchange Values to all agents in a given cluster. Specifically, we consider the case where we want to assign n agents to k clusters, i.e., finding cluster assignments $\mathbf{u} = \{u(0), ..., u(n-1)\}$ with $u(i) \in \{0, \dots, k-1\}$.

We first compute the clustered value $\tilde{v}(C)$ for a coalition of agents $C \subseteq K$ by averaging over all values for agents assigned to the same cluster as

$$\tilde{v}(C) = \frac{1}{\eta} \sum_{m=0}^{n-1} \sum_{\pi \in \Pi_N} v(S_\pi(m)) \cdot \mathbb{1}(u(j) \mid j \in S_\pi(m) = C), \tag{3}$$

where the normalisation constant is defined as $\eta = \sum_{m=0}^{n-1} \sum_{\pi \in \Pi_N} \mathbb{1}(\{u(j) \mid j \in S_\pi(m)\} = C)$.

Given the clustered value function $\tilde{v}(C)$, we denote the Exchange Value (see Definition 4.1) of an agent i as $\tilde{\gamma}_i(\tilde{G})$, with $\tilde{G} = (K, \tilde{v})$.

We propose selecting a cluster assignment $\mathbf{u}$ that maximizes the variance in the Exchange Values (EVs) of all n agents in N. This objective is motivated by the intuition that assigning agents uniformly to clusters minimizes the expected EV of any cluster, resulting in a variance in EVs that is minimized. This occurs because each cluster consists of a combination of agents with both positive and negative contributions, leading to an expected EV of zero.

We first define the optimal cluster assignments $\mathbf{u}^*$ that maximize the variance in EVs as

$$\mathbf{u}^* \in \arg\max_{\mathbf{u}} \mathrm{Var}([\tilde{\gamma}_0(\tilde{G}), \dots, \tilde{\gamma}_{n-1}(\tilde{G})]). \tag{4}$$

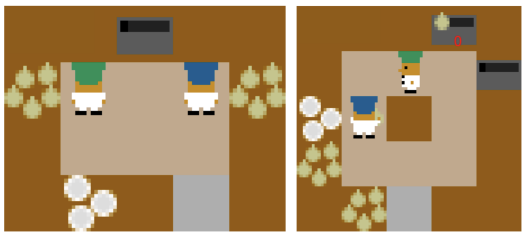

Figure 1: In the Overcooked environments Cramped Room (left) and Coordination Ring (right), agents must cooperate to cook and deliver as many soups as possible within a given time.

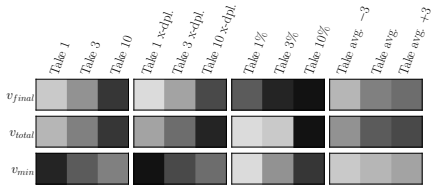

Figure 2: Colour-coded ordering of EVs for agents with varying behaviors in Tragedy of the Commons. The brighter, the higher an agent's contribution to a given value function.

We show (Appendix A.4) that the objective stated in Equation 4 is equivalent to clustering agents by each agent's unobserved trait v_i, under the assumption of an inessential game. In an inessential game, the value of a coalition can be decomposed as $\tilde{v}(C) = \sum_{i \in C} v_i$, with the ith agent contributing a term v_i (the unobserved trait).

It then holds that

$$\begin{aligned} \mathbf{u}^* \in & \arg\max_{\mathbf{u}} \mathrm{Var}([k_0, \ldots, k_{n-1}]), \\ & \text{with } k_i = {}^1\!/_\epsilon \cdot \textstyle\sum_{j \in N} v_j \cdot \mathbb{1}(\mathbf{u}(i) = \mathbf{u}(j)), \end{aligned} \tag{5}$$

where $[k_0, \ldots, k_{n-1}]$ are the centroids of the clusters that each agent $i \in N$ is assigned to, and ϵ is a normalisation constant given as $\sum_{j \in N} \mathbb{1}(\mathbf{u}(i) = \mathbf{u}(j))$.

This objective is therefore equivalent to assigning agents to clusters such that the variance in cluster centroids (centroids computed as the mean of the unobserved traits v_i of all agents assigned to a given cluster) is maximized. In summary, maximizing the variance in EVs is equivalent to maximizing the between-cluster variance (in an inessential game), a common objective in clustering [17], hence allowing to cluster agents by their unobserved trait v_i.

5. Experiments

The environments that we consider only permit certain coalition sizes, hence only Constrained Exchange Values (Def. 4.3) are applicable. We simply refer to them as Exchange Values (EVs) from here onward. As the considered environments are stochastic, we use sampling (as introduced in Section 4.3) to estimate true EVs. We run all experiments for five random seeds and report mean and standard deviation where applicable. For implementation details please refer to the appendix.

Tragedy of the Commons. The Tragedy of the Commons [11] (ToC) refers to a situation where multiple individuals deplete or degrade a shared resource, and is a social-dilemma scenario often used to emphasize the need for proper regulation to avoid the overexploitation of common resources [7, 25]. We model such a scenario as a multi-agent environment in which agents can consume from a common pool of resources x_t, which grows at a fixed rate g at each time step t: $x_{t+1} = \max\left((1+g) \cdot x_t - \sum_i c_{ti},\ 0\right)$, with c_{ti} as the consumption of the ith agent at time t. Hence, if all resources are consumed, none can regrow and no agents can consume more resources. We generate a simulated dataset of interactions of agents with varying strategies for consuming resources (described later in more detail).

Overcooked. Overcooked [4] is a two-player game simulating a cooperative cooking task, where players must prepare as many dishes as possible in a given time, requiring teamwork and coordination. It is a common testbed in multi-agent research for studying collaboration [4, 15, 31]. Within Overcooked, we consider the configurations Cramped Room and Coordination Ring (displayed in Figure 1). For each environment configuration, we generate two datasets by simulating agent behaviors using a near-optimal planning algorithm [4], where we use a parameter λ to determine an agent's behavior, which we refer to as the trait of the agent. For $\lambda = 1$ agents act (near)-optimally, for $\lambda = -1$ agents act adversarially. Each dataset D (defined in Section 3) is generated by $n = 100$ agents, and trajectories τ are of length 400. The adversarial dataset D^{adv} is comprised of 75% agents executing (near)-optimal policies ($\lambda = 1$) and 25% agents executing adversarial policies ($\lambda = -1$), while for the D^{λ} dataset agents were uniformly sampled between $\lambda = -1$ and $\lambda = 1$. We also consider a dataset D^{human} of humans playing the game (provided by [4]), which is fully anonymized, hence each human demonstrator appears only once in the dataset.

5.1. Exchange Values assess an agent's individual contribution to a collective value function

In this section we consider the case when the datasets contain demonstrations of all possible coalitions (groups of agents), which allows us to accurately estimate EVs without missing coalitions.

Tragedy of the Commons. We consider $n = 12$ agents with four different behavior patterns: "Take X" consumes X units at every timestep, "Take X x-dpl" consumes X units if this does not deplete the pool of resources, "Take X%" consumes X% of the available resources, and " TakeAvg" consumes the average of the resources consumed by the other agents at the previous timestep (0 in the first timestep). Each behavior pattern is followed by 3 agents, with $X \in \{1, 3, 10\}$. We simulate ToC for coalitions of size three for 50 time steps, with a starting pool of $x_0 = 200$ resources and a per-timestep growth g of 25%. We evaluate three DVFs: the final pool size (v_{final}), the total resources consumed (v_{total}), and the minimum consumption among agents (v_{min}). These represent different interpretations of social welfare in the game. Figure 2 shows the color-coded ordering of agents by EV under the three different DVFs, suggesting that to maximize alignment with given DVF, agents with lighter colors (higher contributions) should be imitated. The ordering broadly reflects our intuition: taking more resources negatively impacts the EVs, and agents consuming the average of others have less extreme EVs. Taking too few resources reverses this trend for v_{min}.

We now consider imitating the dataset of 12 agents, while learning three imitation policies aligned with each of the three DVFs. We compare (a) imitating the behavior of all agents in the dataset with Behavior Cloning (BC, [28]) and (b) modifying the BC objective to imitate agents with positive contributions to the DVF only, which we refer to as *EV-BC*. Table 1 demonstrates that EV-BC outperforms standard behavior cloning by a large margin. This indicates that considering individual agents' EVs to a given DVF leads to improved respective performance of imitation policies.

Simulated datasets for Overcooked. We now consider the two simulated datasets (D^{adv} and D^{λ}) to evaluate EVs in the Overcooked environment. Note that we consider the human dataset in the next section, as this dataset is incomplete (does not contain observations for all possible coalitions). We consider the value function given for Overcooked as the DVF, i.e. the number of soups prepared by both agents within a trajectory. We compute EVs for all agents in both datasets. Figue 3 (see Appendix) shows that EVs of individual agents are strongly correlated with the trait parameter λ of an individual agent, which approximates how well an agent's behavior is aligned with the DVF.

5.2. Imitating desired behavior with Exchange Values

We now consider all datasets D^{adv}, D^{λ} and D^{human} in both Overcooked environments. We present results for estimating EVs from incomplete data, especially from fully-anonymized human datasets in App. A.2. Note that in the standard Overcooked environment, an adversarial agent is limited to *blocking* the other agent, while in many real-world environments, adversaries are likely to be capable of more diverse (and possibly severe) actions. We introduce a modified version of the Overcooked environment in which agents can take an additional action that lights the kitchen on fire with a predefined probability, resulting in an episode reward of -200; we refer to this environment as *Overcooked+Fire* and evaluate on equivalent datasets D^{adv} and D^{λ}.

We evaluate the performance achieved by agents with respect to the DVF (in this case the environments value function of maximizing the number of soups) when trained with different imitation learning approaches on the different datasets. We use the fully-anonymized datasets and consider the EVs computed from these using our proposed clustering approach. We refer to our method as EV-Behavioral Cloning (*EV-BC*), as we modify the original Behavior Cloning [28] objective to only include trajectories from agents with positive EVs. We also consider the following approaches as relevant baselines: (1) Vanilla BC, where the full dataset is used, (2) the offline multi-agent reinforcement learning algorithm OMAR [27], where we set the reward at the last timestep to the DVF's score for a given trajectory (no per-step reward signal is given by the DVF) and (3) Reward BC, for which we exclude runs with below mean collective score. Note that while our method is specific to individual agents, this baseline considers the score achieved by a group of agents. Table 2 shows that EV-BC clearly outperforms the baseline approaches in both environment configurations, with the margin being more significant in the Overcooked+Fire environments where adversarial agents can take more powerful actions. We further note that EV-BC significantly outperforms baseline approaches on the datasets of human-generated behavior, for which EVs were estimated from a fully-anonymized real-world dataset.

6. Conclusion

Our work presents a method for training AI agents from diverse datasets of human interactions while ensuring that the resulting policy is aligned with a given desired value function. However, it must be noted that quantifying the desired value function poses a challenging research problem in itself. Nonetheless, we expect the potential positive effects of enabling AI designers to align AI agents with relevant value functions to outweigh the potential negative effects which could result from alignment with possibly harmful value functions. Future work may focus on addressing the assumption that individual agents behave similarly across multiple trajectories, by developing methods for a more fine-grained assessment of desired behavior. Additionally, exploring how our framework can more effectively utilize data of undesired behavior is an interesting avenue for further investigation, e.g., developing policies that are constrained to not take undesirable actions.

References

[1] Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T. et al. [2022], 'Training a helpful and harmless assistant with reinforcement learning from human feedback', *arXiv preprint arXiv:2204.05862* .

[2] Beliaev, M., Shih, A., Ermon, S., Sadigh, D. and Pedarsani, R. [2022], Imitation learning by estimating expertise of demonstrators, *in* 'International Conference on Machine Learning', PMLR, pp. 1732–1748.

[3] Cao, Z. and Sadigh, D. [2021], 'Learning from imperfect demonstrations from agents with varying dynamics', *IEEE Robotics and Automation Letters* **6**(3), 5231–5238.

[4] Carroll, M., Shah, R., Ho, M. K., Griffiths, T., Seshia, S., Abbeel, P. and Dragan, A. [2019], 'On the utility of learning about humans for human-ai coordination', *Advances in neural information processing systems* **32**.

[5] Chang, Y.-H., Ho, T. and Kaelbling, L. [2003], 'All learning is local: Multi-agent learning in global reward games', *Advances in neural information processing systems* **16**.

[6] Chen, L., Paleja, R. and Gombolay, M. [2021], Learning from suboptimal demonstration via self-supervised reward regression, *in* 'Conference on robot learning', PMLR, pp. 1262–1277.

[7] Dietz, T., Ostrom, E. and Stern, P. C. [2003], 'The struggle to govern the commons', *science* **302**(5652), 1907–1912.

[8] (FAIR)†, M. F. A. R. D. T., Bakhtin, A., Brown, N., Dinan, E., Farina, G., Flaherty, C., Fried, D., Goff, A., Gray, J., Hu, H. et al. [2022], 'Human-level play in the game of diplomacy by combining language models with strategic reasoning', *Science* **378**(6624), 1067–1074.

[9] Foerster, J., Farquhar, G., Afouras, T., Nardelli, N. and Whiteson, S. [2018], Counterfactual multi-agent policy gradients, *in* 'Proceedings of the AAAI conference on artificial intelligence', Vol. 32.

[10] Glaese, A., McAleese, N., Trębacz, M., Aslanides, J., Firoiu, V., Ewalds, T., Rauh, M., Weidinger, L., Chadwick, M., Thacker, P. et al. [2022], 'Improving alignment of dialogue agents via targeted human judgements', *arXiv preprint arXiv:2209.14375* .

[11] Hardin, G. [1968], 'The tragedy of the commons: the population problem has no technical solution; it requires a fundamental extension in morality.', *science* **162**(3859), 1243–1248.

[12] Hastie, T., Tibshirani, R., Friedman, J. H. and Friedman, J. H. [2009], *The elements of statistical learning: data mining, inference, and prediction*, Vol. 2, Springer.

[13] He, J. Z.-Y., Erickson, Z., Brown, D. S., Raghunathan, A. and Dragan, A. [2023], Learning representations that enable generalization in assistive tasks, *in* 'Conference on Robot Learning', PMLR, pp. 2105–2114.

[14] Heuillet, A., Couthouis, F. and Díaz-Rodríguez, N. [2022], 'Collective explainable ai: Explaining cooperative strategies and agent contribution in multiagent reinforcement learning with Shapley values', *IEEE Computational Intelligence Magazine* **17**(1), 59–71.

[15] Hu, H., Lerer, A., Peysakhovich, A. and Foerster, J. [2020], "other-play" for zero-shot coordination, *in* 'International Conference on Machine Learning', PMLR, pp. 4399–4410.

[16] Jiang, J. and Lu, Z. [2021], 'Offline decentralized multi-agent reinforcement learning', *arXiv preprint arXiv:2108.01832* .

[17] Koch, I. [2013], *Analysis of multivariate and high-dimensional data*, Vol. 32, Cambridge University Press.

[18] Korbak, T., Shi, K., Chen, A., Bhalerao, R., Buckley, C. L., Phang, J., Bowman, S. R. and Perez, E. [2023], 'Pretraining language models with human preferences', *arXiv preprint arXiv:2302.08582* .

[19] Kraft, D. [1988], 'A software package for sequential quadratic programming', *Forschungsbericht-Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt* .

[20] Le, H. M., Yue, Y., Carr, P. and Lucey, P. [2017], Coordinated multi-agent imitation learning, *in* 'International Conference on Machine Learning', PMLR, pp. 1995–2003.

[21] Li, J., Kuang, K., Wang, B., Liu, F., Chen, L., Wu, F. and Xiao, J. [2021], Shapley counterfactual credits for multi-agent reinforcement learning, *in* 'Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining', pp. 934–942.

[22] Littman, M. L. [1994], Markov games as a framework for multi-agent reinforcement learning, *in* 'Machine Learning Proceedings 1994'.

[23] Lloyd, S. P. [1982], 'Least squares quantization in pcm', *IEEE Transactions on Information Theory* **28**(2), 129–137.

[24] Nguyen, D. T., Kumar, A. and Lau, H. C. [2018], 'Credit assignment for collective multiagent rl with global rewards', *Advances in neural information processing systems* **31**.

[25] Ostrom, E. [2009], 'A general framework for analyzing sustainability of social-ecological systems', *Science* **325**(5939), 419–422.

[26] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A. et al. [2022], 'Training language models to follow instructions with human feedback', *Advances in Neural Information Processing Systems* **35**, 27730–27744.

[27] Pan, L., Huang, L., Ma, T. and Xu, H. [2022], Plan better amid conservatism: Offline multi-agent reinforcement learning with actor rectification, *in* 'International Conference on Machine Learning', PMLR, pp. 17221–17237.

[28] Pomerleau, D. A. [1991], 'Efficient Training of Artificial Neural Networks for Autonomous Navigation', *Neural Computation* **3**(1).

[29] Sasaki, F. and Yamashina, R. [2021], Behavioral cloning from noisy demonstrations, *in* 'International Conference on Learning Representations'.

[30] Shapley, L. [1953], 'A value for n-person games', *Contributions to the Theory of Games* pp. 307–317.

[31] Shih, A., Ermon, S. and Sadigh, D. [2022], Conditional imitation learning for multi-agent games, *in* '2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)', IEEE, pp. 166–175.

[32] Song, J., Ren, H., Sadigh, D. and Ermon, S. [2018], 'Multi-agent generative adversarial imitation learning', *Advances in neural information processing systems* **31**.

[33] Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D. and Christiano, P. F. [2020], 'Learning to summarize with human feedback', *Advances in Neural Information Processing Systems* **33**, 3008–3021.

[34] Thorndike, R. [1953], 'Who belongs in the family?', *Psychometrika* **18**(4), 267–276.

[35] Tian, Q., Kuang, K., Liu, F. and Wang, B. [2022], 'Learning from good trajectories in offline multi-agent reinforcement learning', *arXiv preprint arXiv:2211.15612* .

[36] Tseng, W.-C., Wang, T.-H. J., Lin, Y.-C. and Isola, P. [2022], 'Offline multi-agent reinforcement learning with knowledge distillation', *Advances in Neural Information Processing Systems* **35**, 226–237.

[37] Wang, J., Zhang, Y., Gu, Y. and Kim, T.-K. [2022], 'Shaq: Incorporating shapley value theory into multi-agent q-learning', *Advances in Neural Information Processing Systems* **35**, 5941–5954.

[38] Wang, J., Zhang, Y., Kim, T.-K. and Gu, Y. [2020], Shapley q-value: A local reward approach to solve global reward games, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence', Vol. 34, pp. 7285–7292.

[39] Xu, J., Ju, D., Li, M., Boureau, Y.-L., Weston, J. and Dinan, E. [2020], 'Recipes for safety in open-domain chatbots', *arXiv preprint arXiv:2010.07079* .

[40] Yang, M., Levine, S. and Nachum, O. [2021], 'Trail: Near-optimal imitation learning with suboptimal data', *arXiv preprint arXiv:2110.14770* .

[41] Yu, L., Song, J. and Ermon, S. [2019], Multi-agent adversarial inverse reinforcement learning, *in* 'International Conference on Machine Learning', PMLR, pp. 7194–7201.

[42] Zhang, S., Cao, Z., Sadigh, D. and Sui, Y. [2021], 'Confidence-aware imitation learning from demonstrations with varying optimality', *Advances in Neural Information Processing Systems* **34**, 12340–12350.

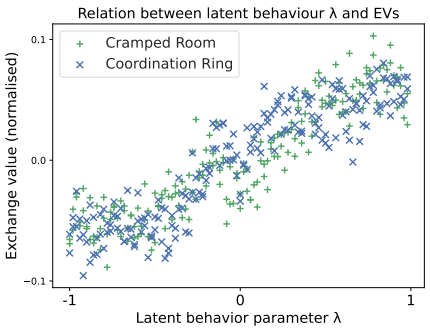

Figure 3: Relation between an agent's trait λ and its EV in Overcooked.

Table 1: Achieved relative alignment with different value functions in ToC, using either BC or EV-BC.

	Method	
	BC	EV-BC (Ours)
v_{final}	23.25%	100%
v_{total}	12.3%	100%
v_{min}	1.3%	100%

A. Appendix

A.1. Appendix to methods

A.1.1. Derivation of clustering objective stated in Eq. 5

Inessential games and EVs. The assumption of an inessential game is commonly made to compute Shapley Values more efficiently[2]. In an inessential game, the value of a coalition is given by the sum of the individual contributions of its members, denoted as $v(C) = \sum_{i \in C} v_i$. We refer to an individual agent's contribution v_i as its unobserved trait. The Exchange Value (see Definition 4.1) of an individual agent i in an inessential game is given as

$$\begin{aligned}\gamma_i(G) &= v_i - \frac{1}{|N|-1} \cdot \sum_{j \in N \setminus \{i\}} v_j \\ &= \left(1 + \frac{1}{|N|-1}\right) \cdot v_i - \frac{1}{|N|-1} \cdot \sum_{j \in N} v_j,\end{aligned}$$

This expression represents the difference between the trait (contribution) of agent i, v_i, and the average trait of all other agents. The second term is independent of i and remains constant across all agents.

Derivation of equivalent clustering objective. We now consider the optimization problem defined by Equation 4, which seeks to find the optimal cluster assignments $\mathbf{u}^*$ that maximize the variance in EVs

$$\mathbf{u}^* = \arg\max_{\mathbf{u}} \text{Var}([\tilde{\gamma}_0(\tilde{G}), \ldots, \tilde{\gamma}_{n-1}(\tilde{G})]).$$

Note that under the concept of a clustered value function (see Equation 3), all agents within a cluster are represented as a single agent. We denote by k_i the latent trait of the agent that represents the agents in cluster i. The value k_i is defined as the average trait of all agents assigned to the cluster, i.e. $k_i = \frac{1}{\epsilon} \cdot \sum_{j \in N} v_j \cdot \mathbb{1}(\mathbf{u}(i) = \mathbf{u}(j))$. Here, the normalization constant is given as $\epsilon = \sum_{j \in N} \mathbb{1}(\mathbf{u}(i) = \mathbf{u}(j))$.

Using the concept of the clustered value function $\tilde{v}$ (see Equation 3), we can express the EV of all agents in cluster i as

$$\tilde{\gamma}_i(\tilde{G}) = \left(1 + \frac{1}{|K|-1}\right) \cdot k_i - \frac{1}{|K|-1} \cdot \sum_{j \in K} k_j.$$

The second term, which is cluster-independent, can be omitted when computing the variance $\text{Var}([\tilde{\gamma}_0(\tilde{G}), \ldots, \tilde{\gamma}_{n-1}(\tilde{G})])$, as the variance is agnostic to a shift in the data distribution. We will omit the scaling factor $\left(1 + \frac{1}{|K|-1}\right)$ from here onwards.

Let n_j denote the number of agents assigned to cluster $j \in K$, with $\sum_{i=0}^{K-1} n_i = N$. By simplifying Equation 4, we obtain:

[2] see, e.g., Covert, I. and Lee, S.I., 2020. Improving kernelshap: Practical shapley value estimation via linear regression. arXiv preprint arXiv:2012.01536.

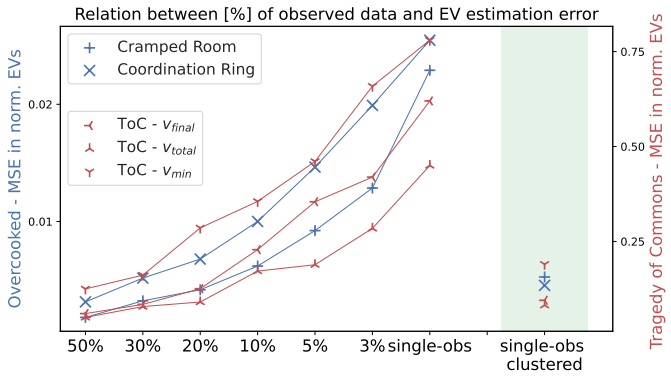

Figure 4: Mean error in estimating EVs with decreasing size of the set of observations. *Single-obs* refers to the fully anonymized case. Estimation error decreases significantly if agents are *clustered* (green-shaded area). In Overcooked we normalize data to have zero mean and unit variance, in ToC using maximum and minimum.

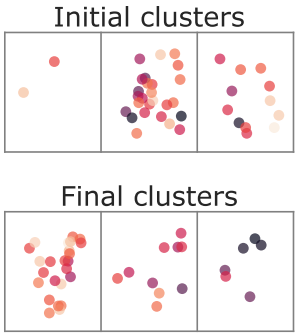

Figure 5: Agents in the three clusters after initial clustering (top) and final clustering (bottom), with agents color-coded by their (unobserved) trait λ_{human}.

$$\text{Var}([\tilde{\gamma}_0(\tilde{G}), \ldots, \tilde{\gamma}_{n-1}(\tilde{G})]) = \sum_{i=0}^{K-1} n_i \cdot \left(k_i - \frac{\sum_{j=0}^{K-1} n_j \cdot k_j}{N} \right)^2.$$

This allows us to express the objective stated in Equation 5 as

$$\mathbf{u}^* = \arg\max_{\mathbf{u}} \text{Var}([k_0, \ldots, k_{n-1}]).$$

A.2. Estimating EVs from incomplete data

Estimation error for different dataset sizes. We now turn to realistic settings with missing data where EVs are estimated by sampling coalitions (see Section 4.3). For both ToC and Overcooked, we compute the estimation error in EVs for different dataset sizes. We first compute ground truth EVs using observations of all coalitions according to Def. 4.3 and then compute the mean estimation error if only a fraction of the possible coalitions is contained in the dataset. As expected, we observe in Figure 4 that the mean estimation error generally increases as the fraction of observed coalitions decreases. We also investigate fully anonymized datasets and find that these have the largest estimation error (see Figure 4 – *single-obs*).

Estimating EVs from fully-anonymised datasets. As estimating EVs from fully-anonymized datasets is a degenerate problem (see App. A.4 for explanation), we combine behavior information, contained in the trajectories τ in D, with the variance maximization objective introduced in Equation 4, and show that this allows clustering agents by their unobserved trait. See Appendix A.4 for an ablation study. Specifically, we first compute initial cluster assignments from agents' low-level behavior, which we then use to imply a soft constraint to the objective of maximizing variance in estimated EVs.

In Overcooked, we first generate an embedding vector per agent by concatenating the agent's empirical action probabilities for the 200 states most frequently observed in a given dataset. This results in a 1000 dimensional embedding that contains information about the agent's behavior only, while discarding other trajectory features which may result from other agents' policies. We then apply dimensionality reduction (PCA, [12]) and k-means clustering [23] (choosing the number of clusters using the ELBOW [34] method), to arrive at behavior-based cluster assignments. In ToC, the behavior-based clusters can be achieved by clustering of action frequencies. We next jointly optimize cluster assignments of individual agents and estimated EVs by maximizing the variance in estimated EVs (see Equation 4), using the previously computed behavior-based cluster assignments to constrain the solution space. We optimize the objective stated in Equaiton 4 using a non-linear constrained optimization solver (SLSQP, [19]), which we initialize with the behavior-based cluster assignments. We further add a small L_2 loss term that penalizes solutions that deviate from the behavior-based clusters (see appendix A.4 for implementation details). We find that this results in a significant decrease in the estimation error of EVs (see Figure 4 – *single-obs clustered*).

Table 2: Resuling performance with respect to the DVF for different imitation learning methods in the Overcooked environments Cramped Room (top) and Coordination Ring (bottom). We find that our approach outperforms the relevant baselines.

	Overcooked			Overcooked+Fire	
Imitation method	$\mathcal{D}^{\lambda}$	$\mathcal{D}^{adv}$	$\mathcal{D}^{\text{human}}$	$\mathcal{D}^{\lambda}$	$\mathcal{D}^{adv}$
vanilla BC [28]	10.8 $\pm$ 2.14	40.8 $\pm$ 12.7	153.34 $\pm$ 11.5	-13.35 $\pm$ 24.5	-20.12 $\pm$ 18.5
reward-BC	54.2 $\pm$ 5.45	64.8 $\pm$ 7.62	163.34 $\pm$ 6.08	24.89 $\pm$ 16.25	0.9 $\pm$ 13.98
Offline RL (OMAR [27])	6.4 $\pm$ 3.2	25.6 $\pm$ 8.9	12.5 $\pm$ 4.5	5.0 $\pm$ 12.5	-3.4 $\pm$ 12.8
EV-BC (ours)	**91.6** $\pm$ 12.07	**104.2** $\pm$ 10.28	**170.89** $\pm$ 6.8	**86.2** $\pm$ 13.02	**98.3** $\pm$ 12.48
vanilla BC [28]	15.43 $\pm$ 4.48	10.4 $\pm$ 6.8	104.89 $\pm$ 12.44	-16.45 $\pm$ 15.6	-40 $\pm$ 14.6
reward-BC	24 $\pm$ 4.69	**14.6** $\pm$ 2.48	102.2 $\pm$ 6.19	-8 $\pm$ 8.59	-51.8 $\pm$ 11.4
Offline RL (OMAR [27])	12.43 $\pm$ 3.35	9.5 $\pm$ 3.5	12.4 $\pm$ 6.0	-0.8 $\pm$ 5.4	-1.2 $\pm$ 5.6
EV-BC (ours)	**30.2** $\pm$ 6.91	12.4 $\pm$ 2.65	**114.89** $\pm$ 5.08	**32.64** $\pm$ 7.14	**12.5** $\pm$ 4.32

Fully-anonymised human-generated dataset for Overcooked. In contrast to the simulated datasets, no ground truth EVs can be computed for the human-generated datasets, as these are fully anonymized. Also, no latent trait λ is given, which could indicate how well a human participant is aligned with the DVF (maximizing the number of soups cooked). To evaluate the goodness of the estimated EVs for the human dataset, we use the keystrokes per second as a proxy for the latent trait (as proposed by Carroll et al. [4]), referring to this value as λ_{human}. We compute EVs for all human participants using the clustering approach described in the previous section. It can be observed in Figure 5 (top row) that the behavior-clusters in Cramped Room (Overcooked) yields a reasonable separation by λ_{human}. The bottom row shows improved cluster assignments after maximizing variance in EVs. Relative to the average within-cluster variance under random cluster assignments, the initial behavior clustering reduces within-cluster variance by 16% and 25% percent in Cramped Room and Coordination ring, respectively, while the final clustering step that maximizes between-cluster variance reduces the within-cluster variance by another 34% and 48% percent, respectively. These findings validate that maximizing variance in EVs allows clustering agents by their unobserved traits, which in this case correspond to λ_{human} (keystrokes per second).

A.3. Overcooked experimental details

Datasets. We generate the simulated datasets using the planning algorithm given in [4][3]. To be able to simulate agents with different behavior (from adversarial to optimal), we first introduc a latent trait parameter, λ, which determines the level of adversarial or optimal actions for a given agent. A value of $\lambda = 1$ represents a policy that always chose the best action with certainty. As λ decreased, agents are more likely to select non-optimal actions. For $\lambda < 0$, we invert the cost function to create agents with adversarial behavior. Notably, we assign a high cost (or low cost when inverted) to occupying the cell next to the counter in the Overcooked environment. Occupying the cell next to the counter enables adversarial agents to block other agents in the execution of their tasks.

For human gameplay datasets, we utilized the raw versions of the Overcooked datasets.[4] These datasets were used as-is, without manual pre-filtering.

Exchange Values. To estimate agents' Exchange Values according to Section 4.3, we used either the full set of all possible coalitions or a fraction of it (see Figure 4 for the relationship between dataset size and EV estimation error). For each observed coalition, we conducted 10 rollouts in the environment and calculated the average score across these rollouts to account for stochasticity in the environment.

Imitation learning. For EV-BC, we modify the standard Behavior Cloning approach [28] by only training on data of agents with a positive estimated EV. As for the BC baseline, we used the complete dataset. In the case of reward-BC, we exclusively utilized data with an above-median return (DVF). For EV-BC, BC, and reward-BC we used the implementation

[3] https://github.com/HumanCompatibleAI/overcooked_ai

[4] https://github.com/HumanCompatibleAI/human_aware_rl/tree/master/human_aware_rl/data/human/anonymized

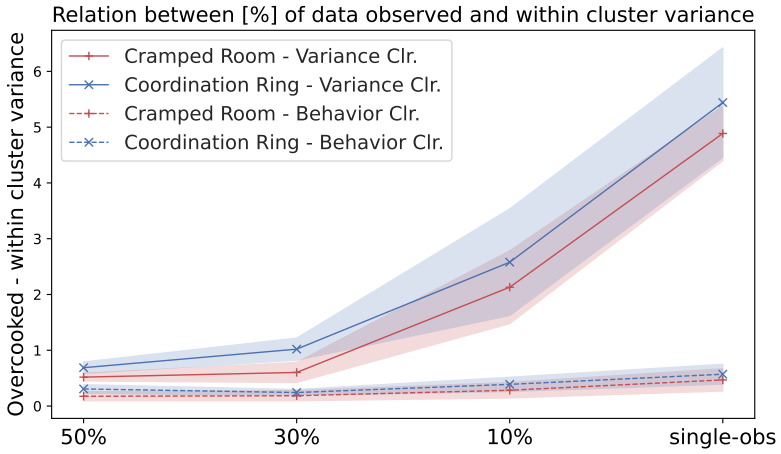

Figure 6: Within-cluster variance in relation to fraction of observations for simulated data in Cramped Room and Coordination Ring (Overcooked). Two clustering methods shown (Behavior clustering and Variance Clustering). In the case of random cluster assignments, the within-cluster variance is 5.11 ± 0.11, while under optimal cluster assignments, the variance is 0.156. See section A.4 for details.

of Behavior Cloning in Overcooked as given by the authors of [4][5]. We implement the offline multi-agent reinforcement learning method OMAR [27] using the author's implementation.[6] For the OMAR baseline, we set the reward at the last timestep to the DVF's score for a given trajectory, as our work assumes that no per-step reward signal is given, in contrast to the standard offline-RL framework. We conducted a hyperparameter sweep for the following parameters: learning rate with options $\{0.01, 0.001, 0.0001\}$, Omar-coe with options $\{0.1, 1, 10\}$, Omar-iters with options $\{1, 3, 10\}$, and Omar-sigma with options $\{1, 2, 3\}$. The best-performing parameters were selected based on the evaluation results.

Implementation of Overcooked+Fire. We introduce an additional adversarial action, "light kitchen on fire," to the environment. To account for this action in the planning algorithm, we assigned it the highest possible cost. Taking this action had a 50% chance of resulting in an episode return of -200, regardless of the other agent's performance.

A.4. Clustering of agents in Overcooked

Degeneracy of credit assignment problem for fully-anonymised datasets. In Definition 4.1, Exchange Values are defined by comparing the value of the coalitions that include a given agent to those that do not. However, when not all coalitions are observed, the exchange value (EV) of an agent can be estimated by sampling coalitions, as discussed in Section 4.3. The estimate is given by

$$\gamma_i(\bar{G}) = \mathbb{E}_{m\sim U(M^+),\pi\sim U(\Pi_{N\setminus\{i\}})}\left[\Gamma^{\bar{G}}_{m,\pi}(i)\right].$$

This can be rewritten as:

$$\gamma_i(\bar{G}) = \mathbb{E}_{m\sim U(M^+),\pi\sim U(\Pi_{N\setminus\{i\}})}\left[v(S_\pi(m-1)\cup\{i\})\right] - \mathbb{E}_{m\sim U(M^+),\pi\sim U(\Pi_{N\setminus\{i\}})}\left[v(S_\pi(m))\right].$$

Here, the first term estimates the value of a coalition that includes agent i, while the second term estimates the value of a coalition that does not include agent i. In the case of a fully-anonymized dataset, each agent is observed only once as part of one multi-agent trajectory. Consequently, the first term must be estimated from a single sample, representing the DVF's score observed for the one appearance of agent i. This leads to a high variance in the estimation of EVs and results in the degeneracy of the problem in attributing contributions to individual agents. Specifically, due to the fact that a single

[5] https://github.com/HumanCompatibleAI/overcooked_ai/tree/master/src/human_aware_rl/imitation

[6] https://github.com/ling-pan/OMAR

score is shared among all agents within one multi-agent trajectory, all agents in that multi-agent trajectory (coalition) are assigned equal EVs. As a result, in the case of a fully-anonymized dataset, it becomes impossible to assign credit specifically to individual agents within a coalition. To mitigate this problem, we propose clustering agents, which we describe in Section A.2. Through clustering, we can assign equal EVs to agents that are similar, in contrast to assigning equal EVs to all agents in a given coalition.

Behavior clustering. The behavior clustering process in the Overcooked environment involves the following steps. Initially, we identify the 200 states that are most frequently visited by all agents in a given set of observations. As the action space in Overcooked is relatively small (<7 actions), we calculate the empirical action distribution for each state for every agent. These 200 action distributions are then concatenated to form a behavior embedding for each agent. To reduce the dimensionality of the embedding, we apply Principal Component Analysis (PCA), transforming the initial embedding space into three dimensions. Subsequently, we employ the k-means clustering algorithm to assign agents to behavior clusters. The number of clusters (7 for Overcooked) is determined using the ELBOW method [34], while linear kernels are utilized for both PCA and k-means. It is noteworthy that the results are found to be relatively insensitive to the parameters used in the dimensionality reduction and clustering steps, thus standard implementations are employed for both methods. Importantly, this clustering procedure focuses exclusively on the observed behavior of agents, specifically the actions taken in specific states, and is independent of the scores assigned to trajectories by the DVF.

Variance clustering. In contrast to behavior clustering, variance clustering (see Section 4.4) focuses solely on the scores assigned to trajectories by the DVF and disregards agent behavior. The objective of variance clustering is to maximize the variance of the clustered EVs, as stated in Equation 4. To optimize this objective, we utilize the SLSQP non-linear constrained optimization solver [19].

We employ soft cluster assignments and enforce constraints to ensure that the total probability is equal to 1 for each agent. The solver is initialized with a uniform distribution and runs until convergence or for a maximum of 100 steps. Given that the optimization problem may have local minima, we perform 500 random initializations and optimizations, selecting the solution with the lowest loss (i.e. the highest variance).

As described in Section A.2, behavior clustering (which utilizes behavior information but disregards DVF scores) and variance clustering (which utilizes DVF scores but disregards behavior information) can be combined. To accomplish this, we initialize the SLSQP solver with the cluster assignments obtained from behavior clustering and introduce a small loss term in the objective function of Equation 4. This additional $l2$ loss term, weighted by 0.1 (selected in a small sensitivity analysis), penalizes deviations from the behavior clusters. Similar to before, we perform 500 iterations while introducing a small amount of noise to the initial cluster assignments at each step. The solution with the highest variance is then selected.

Ablation study. In this section, we present an ablation study to examine the impact of different components in the clustering approach discussed in Section A.2. We proposed two sequential clustering methods: behavior clustering and variance clustering. This ablation study investigates the performance of both clustering steps when performed independently, also under the consideration of the fraction of the data that is observed. We assess performance as the within-clsuter variance in the unobserved agent-specific latent trait variable λ, where lower within-cluster variance indicates higher performance. It is important to note that λ is solely used for evaluating the clustering steps and not utilized during the clustering process.

The results of the ablation study are depicted in Figure 6. Providing context, the within-cluster variance under random cluster assignments is 5.11 ± 0.11, while the within-cluster under optimal cluster assignments is 0.156.

We first discuss variance clustering. Clustering agents based on EVs (variance clustering) as introduced in Seciton 4.4 generally leads to a significant decrease in within-cluster variance in the unobserved variable λ. More specifically, the proposed variance clustering approach (when 50% of data is observed), results in a $\sim 89\%$ reduction of the within-cluster variance in λ, which validates the approach of clustering agents by their unobserved traits by maximizing the variance in estimated EVs. However, we observe in Figure 6 that, as the fraction of observed data decreases, the within-cluster variance increases, indicating a decrease in the quality of clustering. The highest within-cluster variance is observed when using only a single observation ('single-obs'), which corresponds to a fully-anonymized dataset. This finding is consistent with the fact that a fully-anonymized dataset presents a degenerate credit assignment problem, as discussed earlier in Section A.4.

We now discuss behavior clustering. Figure 6 shows that behavior clustering generally results in a very low within-cluster variance. However, it is important to note that these results may not directly translate to real-world data, as the ablation

study uses simulated trajectories. Note that such an ablation study cannot be conducted for the given real-wold human datasets, as these are fully anonymized. In Section A.2, we demonstrate that behavior clustering alone may not be sufficient for fully-anonymized real-world human datasets. Instead, a combination of both behavior clustering and variance clustering yields superior results.

A.5. Tragedy of the Commons experiments

The Tragedy of the Commons (ToC) scenario involves 12 agents. Each agent exhibits one of four behavior patterns: "Take X", which consumes X units at each time step; "Take X x-dpl". which consumes X units as long as it does not deplete the pool of resources; "Take X%", which consumes X% of the available resources; and "TakeAvg", which consumes the average of the resources consumed by the other agents in the previous time step (0 in the first time step).

For each behavior pattern, we consider three agents, with X values selected from the set $1, 3, 10$. To generate a population of agents, we replicate each agent type 5 times. We simulate the ToC scenario for coalitions of size three, spanning 50 time steps. The initial pool of resources is $x_0 = 200$, and the resources grow at a rate of 25% per time step.

Due to the continuous nature of the state and action spaces in ToC, we first discretize both and then apply the same clustering methods used in the Overcooked scenario. We proceed by computing Exchange Values (EVs) for all agents as done in Overcooked (see Figure 4 for results).

Implementation of Behavior Cloning (BC) and EV-BC. We implement imitation policies by replicating the averaged action distributions in the discretized states. In Table 1, we present the results obtained using the full dataset (averaging over all agents), as well as using data solely from agents with positive EVs. For each DVF in Table 1 we report the score relative to the maximum achieved score across both methods (BC and EV-BC). In all cases, BC achieves a fraction of the score achieved by EV-BC.

A.6. Computational demand and reproducibility

We used an Intel(R) Xeon(R) Silver 4116 CPU and an NVIDIA GeForce GTX 1080 Ti (only for training BC, EV-BC, reward-BC, and OMAR policies). In Overcooked, generating a dataset took a maximum of three hours, and estimating EVs from a given dataset takes a few seconds. Behavior clustering consumes a couple of minutes, while Variance clustering took up to two hours per configuration (note that it is run 500 times). Training of the BC, reward-BC, and EV-BC policies took no more than 30 minutes (using a GPU), while the OMAR baseline was trained for up to 2 hours. In Tragedy of Commons, each rollout only consumes a couple of seconds. Clustering times were comparable to those in Overcooked. Computing imitation policies is similarly only a matter of a few minutes.