MA-UITD: OFFLINE MULTI-AGENT REWARD INFERENCE FOR CHARACTERIZING CONTINUOUS SOCIAL BEHAVIOR

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent inverse reinforcement learning offers a powerful framework not only for imitation learning but also for understanding the neural computations supporting social interactions and decision-making. Inferring rewards from interacting agents is challenging because observed choices reflect both personal preferences and expectations about others. Existing methods often require additional environment interaction, repeated policy optimization as an inner loop, or restrictive assumptions about environment dynamics and action spaces, limiting their applicability to fixed behavioral datasets. We introduce Multi-Agent Unilateral Inverse Temporal Difference Learning (MA-UITD), an offline method for inferring explicit agent-specific rewards from joint demonstrations in continuous-state environments with either discrete or bounded continuous actions. Its core idea is to represent the long-term consequences of each agent's own actions through belief-conditioned successor features that average over other agents' behavior. Shared reward features and agent-specific preference vectors connect these unilateral values to the target reward function, while inverse temporal-difference consistency and noise-contrastive behavioral learning enable inference without known dynamics or an inner policy-optimization loop. Across simulated cooperative, competitive, and general-sum tasks, MA-UITD outperforms baselines in discrete-action settings and infers behaviorally compatible reward profiles in continuous-action settings. When applied to experimentally collected social-foraging trajectories, MA-UITD inferred distinct spatial and interaction-dependent reward structures for leaders and followers, providing a quantitative characterization of latent behavioral objectives that may be relevant for understanding the neural computations underlying social behavior.

1 INTRODUCTION

A central goal of behavioral science and neuroscience is to understand the objectives that organize individual behavior. Although accurate action prediction is an important benchmark, an *explicit reward* function provides a more interpretable computational account by specifying which outcomes an animal values, how it trades off competing outcomes, and how these valuations change with experience. Subjective-value signals have been identified in orbitofrontal cortex and across a broader valuation network involving ventromedial prefrontal cortex and ventral striatum (Padoa-Schioppa & Assad, 2006; Bartra et al., 2013), whereas phasic activity of midbrain dopamine neurons has been linked to quantitative reward-prediction errors and corresponding distribution (Schultz et al., 1997; Bayer & Glimcher, 2005; Dabney et al., 2020). Consequently, inferred rewards—and the values and prediction errors they imply—can serve as trial-by-trial computational variables for model-based analyses of neural recordings, enabling tests of latent-computation hypotheses beyond action prediction alone (O'Doherty et al., 2007).

Inverse reinforcement learning (IRL) provides a principled framework for recovering such latent objectives by inferring reward functions under which observed behavior is likely. In single-agent settings, IRL has been used to recover both stable and dynamically evolving goals from behavior (Ziebart et al., 2008; Ashwood et al., 2022; Wu et al., 2025; Ke et al., 2025; Wang et al., 2025). Model-derived reward and value representations have also begun to provide computational vari-

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

ables for characterizing neural activity and examining how valuation-related computations may be altered in neuropsychiatric disorders (Bromberg-Martin et al., 2024; Moneta et al., 2024). However, conventional IRL was developed primarily for single-agent settings and treated each individual as an isolated decision-maker, making it difficult to distinguish intrinsic preferences from socially contingent choices that depend on other individuals.

Inferring these objectives becomes substantially more challenging when individuals interact. In social settings, an animal's choice depends on both its own preferences and its expectations about others. Approaching a shared resource, for example, may reflect one's own resource preference and anticipated interaction with other agents such as cooperation and competition. Similar observed actions can consequently arise from different incentives. Computational models have framed such social inference as IRL in a partially observable Markov decision process (POMDP), where a focal agent acts according to beliefs about an incompletely observed environment (Jara-Ettinger, 2019; Baker et al., 2017; Velez-Ginorio et al., 2017). Although this formulation captures uncertainty and false beliefs, other individuals are typically incorporated into the focal agent's environment rather than represented as autonomous decision-makers with their own actions, policies, and rewards. By contrast, a multi-agent MDP, or Markov game, explicitly represents multiple decision-makers whose joint actions determine state transitions and whose policies are mutually interdependent. Multi-agent inverse reinforcement learning (MAIRL) builds on this formulation to infer agent-specific rewards while accounting for how each agent anticipates and responds to the behavior of others (Yu et al., 2019; Tian et al., 2021; Neumeyer et al., 2021; Chen et al., 2026). MAIRL thus offers a systematic framework for separating what each individual values from what it expects others to do, providing interpretable computational variables for studying cooperation, competition, and general-sum interactions.

Applying MAIRL to animal behavior imposes several practical constraints: experiments typically yield a fixed dataset of joint trajectories without reward labels, known transition dynamics, or opportunities to collect additional behavior under a learned policy, and the recorded states and actions are often continuous. Existing methods address parts of this setting: Multi-Agent Adversarial Inverse Reinforcement Learning(MA-AIRL) combines reward estimation with adversarial policy optimization, requiring repeated policy updates and environment interaction during training (Yu et al., 2019). Multi-Agent Marginal Q-Learning(MAMQL) avoids adversarial training by learning through opponent-marginalized values, using discrete actions and additional policy rollouts in its reported training procedure (Haynam et al., 2025). These constraints motivate a fully offline approach that infers explicit agent-specific rewards directly from fixed joint demonstrations, without known dynamics or an inner policy-optimization loop, while supporting *continuous behavior*.

We introduce Multi-Agent Unilateral Inverse Temporal Difference Learning (MA-UITD), an offline method for inferring explicit agent-specific rewards from joint demonstrations. We focus on arrival-state rewards that describe the value individuals assign to attained environmental and social states. The inferred rewards provide interpretable explanations of behavior while supporting expert-level behavioral imitation. Building on successor features and inverse temporal-difference learning (Filos et al., 2021), our main contributions are:

- We introduce belief-conditioned successor features (BCSF), which marginalize over other agents' anticipated behavior to represent unilateral long-term values and recover interpretable agent-specific rewards.
- We develop an offline learning procedure that combines inverse temporal-difference consistency with noise-contrastive behavioral learning, requires no known dynamics or policy-optimization loop, and supports *continuous states* with either discrete or *continuous actions*.
- Across cooperative, competitive, and general-sum simulations, MA-UITD outperforms discrete-action baselines and captures task-relevant reward structure in continuous-action settings.
- Applied to animal social-interaction data, MA-UITD reveals distinct spatial and interaction-dependent incentives underlying leader–follower dynamics.

2 Related Work

Multi-agent inverse reward learning. In multi-agent settings, MA-AIRL (Yu et al., 2019) couples agent-specific reward estimation to an adversarial discriminator and policy optimization. Such

adversarial approaches involve an inner policy loop and outer loop, which could be computationally expensive and unstable. MAMQL (Haynam et al., 2025) instead learns opponent-marginalized reward through a non-adversarial inverse soft-Q objective; however, it assumes a discrete action set and obtains explicit rewards through a separate reconstruction loss, with its reported training procedure drawing additional policy rollouts. Other work models strategic beliefs through latent quantal level-k reasoning (Tian et al., 2021), but requires recursive value iteration and identifies continuous state and action spaces as a limitation. GS-CIOC (Neumeyer et al., 2021) supports non-adversarial reward inference with continuous states and actions, yet presumes known dynamics and relies on local linear-quadratic approximations around demonstrations under a specified equilibrium model.

Successor-feature-based inverse reward learning. A complementary approach connects explicit rewards to long-term values through a shared feature representation. Filos et al. (2021) introduce inverse temporal-difference (ITD) learning, which jointly estimates shared cumulants, demonstrator-specific successor features, and preference vectors from fixed demonstrations without reward labels. By combining behavioral likelihood fitting with successor-feature Bellman consistency, ITD yields explicit rewards from the learned cumulants and preferences. However, its formal setup considers different reward tasks within a common controlled Markov process with discrete actions, without an explicit Markov-game formulation of strategic interactions.

Offline reward inference for animal behavior. These considerations are particularly important for studies of animal behavior, where joint trajectories can be recorded experimentally but constructing a faithful interactive simulator is often difficult because the underlying behavioral dynamics are only partially known (Fujii et al., 2025). In such settings, methods that require repeated environment interaction, additional policy rollouts, or an accurate model of the dynamics may therefore be impractical. Offline reward-inference methods can recover explicit rewards from fixed demonstrations without known dynamics or additional environment interaction, but are formulated for single-agent settings with discrete actions (Chan & Schaar, 2021). This motivates methods that can infer explicit, agent-specific rewards directly from fixed datasets of continuous joint trajectories in multi-agent systems.

Building on these, our method learns belief-conditioned successor features directly from a fixed dataset of joint transitions and uses noise-contrastive estimation to avoid the intractable continuous-action normalizer. It thereby recovers an explicit agent-specific reward profile in continuous state and action spaces, without adversarial optimization, additional environment interaction, or a forward policy-optimization loop; moreover, equilibrium and finite-depth reasoning enter as restrictions on a common predictive belief rather than as hard-coded solution concepts.

3 Problem Formulation

Markov game and offline demonstrations. We consider an n-agent discounted Markov game without known reward functions $\mathcal{G}^{-R} = (\mathcal{N}, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^n, P, p_0, \gamma)$, where $\mathcal{N} = \{1, \dots, n\}$ is the set of agents, $\mathcal{S}$ is the joint state space, $\mathcal{A}_i$ is the action space of agent i, $\mathcal{A} = \prod_{i=1}^n \mathcal{A}_i$ is the joint action space, $P(s' \mid s, \boldsymbol{a})$ denotes the transition dynamics, p_0 is the initial-state distribution, and $\gamma \in [0, 1)$ is the discount factor. We allow each $\mathcal{A}_i$ to be either discrete or a bounded continuous subset of $\mathbb{R}^{d_i}$.

At time t, all agents simultaneously choose a joint action $\boldsymbol{a}_t = (a_{1,t}, \dots, a_{n,t})$, after which the environment transitions according to $s_{t+1} \sim P(\cdot \mid s_t, \boldsymbol{a}_t)$. We are given only an offline dataset of joint expert trajectories

$$\mathcal{D} = \left\{\tau^{(m)}\right\}_{m=1}^{M}, \qquad \tau^{(m)} = \left(s_0^{(m)}, \boldsymbol{a}_0^{(m)}, s_1^{(m)}, \boldsymbol{a}_1^{(m)} \dots, s_{T_m}^{(m)}, \boldsymbol{a}_{T_m}^{(m)}\right),$$

without observing the rewards that generated the demonstrated behavior. No additional interaction with the environment is assumed during inverse learning.

Arrival-state rewards. Motivated by behavioral and neuroscience settings in which the inferred reward is interpreted as the value assigned to attained outcomes or configurations, we focus on agent-specific arrival-state rewards. That is, the reward received by agent i following a transition

$(s_t, \mathbf{a}_t, s_{t+1})$ is assumed to take the form

$$R_i(s_t, \boldsymbol{a}_t, s_{t+1}) = r_i(s_{t+1}), \qquad r_i : \mathcal{S} \to \mathbb{R}.$$

Because s is the joint environmental state, $r_i(s)$ may still encode both individual preferences and interactions among agents. Thus, this assumption restricts where reward is assigned—to the resulting state rather than directly to the executed action—without requiring agents to share the same reward function. When deterministic choices produce distinguishable outcome states, action-contingent rewards can be represented equivalently as arrival-state rewards. This includes common cognitive tasks such as multi-armed bandits and the prisoner's dilemma when the outcome state records the selected action or joint action.

Reward compatibility rather than reward identification. Inverse reinforcement learning is generally non-identifiable, and this ambiguity is further compounded in multi-agent settings. In particular, rewards in general-sum Markov games are not uniquely identifiable from demonstrations without additional assumptions (Freihaut & Ramponi, 2025). We therefore seek a representative agent-specific reward profile $\widehat{R} = (\hat{r}_1, \ldots, \hat{r}_n)$ within the arrival-state reward class, rather than claim recovery of the unique ground truth. We assess compatibility operationally: under agent i's predictive belief $\boldsymbol{\mu}_{i,-i}$, the reward-induced belief-conditioned score should assign higher relative likelihood to demonstrated actions than to counterfactual alternatives. Equilibrium and finite-depth reasoning enter only through the choice of this belief, rather than as imposed solution concepts.

4 Multi-Agent Unilateral Inverse Temporal Difference Learning (MA-UITD)

We introduce MA-UITD, an offline method for inferring agent-specific rewards from fixed joint demonstrations. It involves three key components: a shared reward feature map Φ, an agent-specific preference vector w_i, and agent-specific belief-conditioned successor features (BCSFs). BCSFs represent the long-term consequences of an agent's own action after averaging over its prediction of the other agents' behavior. This separates strategic assumptions, encoded by the predictive belief, from reward preferences, encoded by w_i.

Learning combines two complementary objectives. A contrastive behavioral loss makes the inferred values assign greater relative likelihood to demonstrated actions than to sampled alternatives, while a unilateral inverse temporal-difference loss enforces Bellman consistency between the BCSFs and the learned reward features. Both objectives are estimated from the fixed dataset, without a transition model, additional environment interaction, or an inner policy-optimization loop. We first define the BCSF representation and then present the offline learning procedure.

4.1 Belief-Conditioned Successor Features (BCSF)

For a stationary joint policy $\boldsymbol{\pi}$, define agent i's joint-action value as

$$Q_i^{\boldsymbol{\pi}}(s, \boldsymbol{a}) \triangleq \mathbb{E}_{\boldsymbol{\pi}, P}\left[\sum_{t=0}^{\infty} \gamma^t r_i(s_{t+1}) \,\middle|\, s_0 = s, \boldsymbol{a}_0 = \boldsymbol{a}\right]. \tag{1}$$

This value conditions on the complete joint action $\boldsymbol{a} = (a_i, \boldsymbol{a}_{-i})$. The initial joint action is fixed, while subsequent joint actions are generated by $\boldsymbol{\pi}$ and states evolve according to P. In our setting, the demonstrations consist of trajectories generated by an unknown stationary joint policy $\boldsymbol{\pi}^D$.

The joint-action value $Q_i^{\boldsymbol{\pi}}(s, a_i, \boldsymbol{a}_{-i})$ evaluates a fully specified joint action. In a simultaneous-move interaction, however, agent i must select a_i without observing the other agents' current actions. Its policy therefore cannot respond directly to the realized $\boldsymbol{a}_{-i}$. Instead, evaluating a_i requires a belief about the actions of the other agents. Let $\boldsymbol{\mu}_{i,-i}(\boldsymbol{a}_{-i} \mid s)$ denote its belief about the other agents' actions, and let $\boldsymbol{\pi}^{(i)} = (\pi_i, \boldsymbol{\mu}_{i,-i})$ denote its perceived continuation profile. The value relevant to its choice is therefore

$$\bar{Q}_i^{\pi_i, \boldsymbol{\mu}_{i,-i}}(s, a_i) \triangleq \mathbb{E}_{\boldsymbol{a}_{-i} \sim \boldsymbol{\mu}_{i,-i}(\cdot|s)}\left[Q_i^{\boldsymbol{\pi}^{(i)}}(s, a_i, \boldsymbol{a}_{-i})\right]. \tag{2}$$

We assume an arrival-state linear reward representation,

$$r_i(s_{t+1}) = \Phi(s_{t+1})^\top w_i. \tag{3}$$

The corresponding joint-action successor features are

$$\Psi_i^{\boldsymbol{\pi}^{(i)}}(s, \boldsymbol{a}) \triangleq \mathbb{E}_{\boldsymbol{\pi}^{(i)}, P}\left[\sum_{t=0}^{\infty} \gamma^t \Phi(s_{t+1}) \,\middle|\, s_0 = s, \boldsymbol{a}_0 = \boldsymbol{a}\right], \tag{4}$$

so that $Q_i^{\boldsymbol{\pi}^{(i)}}(s, \boldsymbol{a}) = \Psi_i^{\boldsymbol{\pi}^{(i)}}(s, \boldsymbol{a})^\top w_i$. Marginalizing these successor features over the agent's belief gives the *belief-conditioned successor features* (BCSFs):

$$\bar{\Psi}_i^{\pi_i, \boldsymbol{\mu}_{i,-i}}(s, a_i) \triangleq \mathbb{E}_{\boldsymbol{a}_{-i} \sim \boldsymbol{\mu}_{i,-i}(\cdot|s)}\left[\Psi_i^{\boldsymbol{\pi}^{(i)}}(s, a_i, \boldsymbol{a}_{-i})\right]. \tag{5}$$

By linearity,

$$\bar{Q}_i^{\pi_i, \boldsymbol{\mu}_{i,-i}}(s, a_i) = \bar{\Psi}_i^{\pi_i, \boldsymbol{\mu}_{i,-i}}(s, a_i)^\top w_i. \tag{6}$$

We model agent i's policy as a logit response to this unilateral value:

$$\pi_i(a_i \mid s) = \frac{\exp\left(\lambda_i \bar{\Psi}_i^{\pi_i, \boldsymbol{\mu}_{i,-i}}(s, a_i)^\top w_i\right)}{\displaystyle\int_{\mathcal{A}_i} \exp\left(\lambda_i \bar{\Psi}_i^{\pi_i, \boldsymbol{\mu}_{i,-i}}(s, \tilde{a}_i)^\top w_i\right) d\tilde{a}_i}. \tag{7}$$

For discrete action spaces, the integral is replaced by a sum over $\mathcal{A}_i$.

The BCSFs satisfy the Bellman recursion

$$\bar{\Psi}_i^{\pi_i, \boldsymbol{\mu}_{i,-i}}(s, a_i) = \mathbb{E}_{\substack{\boldsymbol{a}_{-i} \sim \boldsymbol{\mu}_{i,-i}(\cdot|s), \\ s' \sim P(\cdot|s, a_i, \boldsymbol{a}_{-i}), \\ a_i' \sim \pi_i(\cdot|s')}}\left[\Phi(s') + \gamma \bar{\Psi}_i^{\pi_i, \boldsymbol{\mu}_{i,-i}}(s', a_i')\right]. \tag{8}$$

This recursion integrates over the other agents' current actions, the next state, and agent i's continuation action. It is the basis of the offline temporal-difference objective introduced below.

The belief $\boldsymbol{\mu}_{i,-i}$ determines the strategic interpretation of the representation. For a logit quantal-response equilibrium (LQRE) (McKelvey & Palfrey, 1995), there is a common policy profile $\boldsymbol{\pi}^\star$ such that, for every agent,

$$\boldsymbol{\mu}_{i,-i} = \boldsymbol{\pi}^\star_{-i} = \prod_{j \neq i} \pi_j^\star, \qquad \pi_i = \pi_i^\star. \tag{9}$$

All perceived continuation profiles then coincide with $\boldsymbol{\pi}^\star$, and Equation (7) holds simultaneously for every agent. Under quantal level-k reasoning, the same representation instead uses $\boldsymbol{\mu}_{i,-i}^{(k)} = \boldsymbol{\pi}_{-i}^{(k-1)}$. Thus, equilibrium and finite-depth reasoning correspond to different restrictions on the predictive belief rather than different reward representations.

4.2 Offline Reward Inference with MA-UITD

Given demonstrations $\mathcal{D}$, we parameterize the shared reward feature map as $\Phi(s; \boldsymbol{\theta}^\Phi)$, agent i's BCSFs as $\bar{\Psi}_i(s, a_i; \boldsymbol{\eta}_i)$, and its preference vector as w_i. For offline estimation, we take the predictive belief to be calibrated to the demonstrated conditional behavior of the other agents $\boldsymbol{\mu}_{i,-i}(\boldsymbol{a}_{-i} \mid s) \approx \boldsymbol{\pi}_{-i}^D(\boldsymbol{a}_{-i} \mid s)$. Observed joint transitions then provide Monte Carlo samples of the expectations in Equation (8). We fit the parameterized representation using a contrastive behavioral objective and a BCSF temporal-difference objective.

4.2.1 Contrastive Behavioral Learning

The normalizing integral in Equation (7) is generally intractable for continuous actions. We therefore fit the unnormalized BCSF scores using noise-contrastive estimation (NCE) (Gutmann & Hyvärinen, 2010). At a demonstrated state s, let a_i^+ denote agent i's observed action and sample K alternative actions $a_{i,1:K}^- \sim \nu_i(\cdot \mid s)$ from a declared proposal distribution over $\mathcal{A}_i$. Their scores are $q_i^+ = \bar{\Psi}_i(s, a_i^+; \boldsymbol{\eta}_i)^\top w_i$, $q_{i,k}^- = \bar{\Psi}_i(s, a_{i,k}^-; \boldsymbol{\eta}_i)^\top w_i$.

We minimize the contrastive loss

$$\mathcal{L}_i^{\mathrm{NCE}}(\boldsymbol{\eta}_i, w_i) \triangleq -\mathbb{E}_{\substack{(s,a_i^+)\sim\mathcal{D},\\ a_{i,1:K}^-\sim\nu_i(\cdot|s)}} \left[\log \frac{\exp(q_i^+/\tau_{\mathrm{NCE}})}{\exp(q_i^+/\tau_{\mathrm{NCE}}) + \sum_{k=1}^K \exp(q_{i,k}^-/\tau_{\mathrm{NCE}})}\right]. \tag{10}$$

where $\tau_{\mathrm{NCE}} > 0$ controls the assumed stochasticity of the demonstrated behavior and absorbs the rationality scale λ_i in Equation (7). A lower temperature emphasizes the highest-valued candidate, whereas a higher temperature represents noisier action selection. This objective compares only unilateral alternatives for agent i; the other agents' behavior remains marginalized through $\bar{\Psi}_i$.

4.2.2 Unilateral inverse temporal-difference learning

Inverse temporal-difference learning connects reward features to successor features through their Bellman recursion (Filos et al., 2021). In our setting, the recursion must be enforced on BCSFs because these are the unilateral quantities that determine the behavioral scores in Equation (10).

Consider consecutive demonstrated transitions $(s_t, \boldsymbol{a}_t, s_{t+1}, \boldsymbol{a}_{t+1}) \sim \mathcal{D}$. Under Section 4.2, the observed $\boldsymbol{a}_{-i,t}$ samples the current belief, s_{t+1} samples the unknown transition kernel, and $a_{i,t+1}$ samples agent i's continuation policy. The one-step target

$$y_{i,t} = \Phi(s_{t+1}; \boldsymbol{\theta}^\Phi) + \gamma\, \mathrm{stopgrad}\big(\bar{\Psi}_i(s_{t+1}, a_{i,t+1}; \tilde{\boldsymbol{\eta}}_i)\big) \tag{11}$$

is therefore a Monte Carlo estimate of the right-hand side of Equation (8). Here $\tilde{\boldsymbol{\eta}}_i$ denotes target-network parameters, which are treated as fixed when differentiating the loss. We minimize

$$\mathcal{L}_i^{\mathrm{UITD}}(\boldsymbol{\theta}^\Phi, \boldsymbol{\eta}_i) \triangleq \mathbb{E}_{(s_t, \boldsymbol{a}_t, s_{t+1}, \boldsymbol{a}_{t+1})\sim\mathcal{D}} \left[\left\| \bar{\Psi}_i(s_t, a_{i,t}; \boldsymbol{\eta}_i) - y_{i,t} \right\|^2\right]. \tag{12}$$

This sample-based objective implicitly averages over the other agents' actions and the environment transition, so neither the transition kernel nor an explicit opponent model must be estimated.

4.2.3 Optimization and reward recovery

Training alternates between the two objectives. See Algorithm 1 for a detailed implementation. The NCE step updates $\boldsymbol{\eta}_i$ and w_i to fit the demonstrated action preferences, with an ℓ_1 penalty $\lambda_w \|w_i\|_1$. The UITD step updates $\boldsymbol{\theta}^\Phi$ and every $\boldsymbol{\eta}_i$ to enforce temporal consistency, followed by an update of the target parameters $\tilde{\boldsymbol{\eta}}_i$. After convergence, the agent-specific rewards are recovered directly as $\widehat{r}_i(s) = \Phi(s; \boldsymbol{\theta}^\Phi)^\top w_i$. Because each agent has its own w_i, the same representation applies to cooperative, competitive, and general-sum games.

5 Results

We evaluate MA-UITD in three simulated multi-agent environments spanning cooperative, competitive, and general-sum interactions. Our experiments examine 1) whether MA-UITD reproduces expert-level behavior and 2) recovers interpretable rewards compatible with the demonstrations. We first compare MA-UITD with baselines in discrete-action settings and then evaluate it in continuous-action settings.

5.1 Experimental setup

Environments. We evaluate three games: (1) cooperative *Simple Spread* from MPE2 (Lowe et al., 2017), where two agents cooperate to cover two landmarks; (2) competitive *Simple Push*, where a good agent attempts to reach a target landmark while an adversary tries to prevent it from doing so through physical interaction; and (3) general-sum *Gems*, adapted from Haynam et al. (2025). In *Gems*, Red and Blue maximize individual returns by collecting their respective private gems for 1 reward or jointly collecting a purple gem for 6, after which the episode terminates. Agents choose among nine directional actions in the discrete setting or bounded two-dimensional displacements in the continuous setting, with an 80-step horizon.

Baselines. We evaluate MA-UITD against two baselines. The first is MA-AIRL (Yu et al., 2019), an adversarial learning-based multi-agent IRL method. Then, we benchmark against MAMQL (Haynam

et al., 2025), an IQ-Learn-style MAIRL algorithm that similarly marginalizes the joint-action value function over the opponents' continuation policies but is designed only for discrete action spaces.

5.2 Policy and Reward Recovery

Policy imitation. MA-UITD achieved higher recovered-policy returns than both baselines in *Simple Spread* and for both roles in *Simple Push*, while most closely approaching expert performance in *Gems* (Table 1). In *Gems*, MA-UITD and MAMQL recovered both cooperative and competitive behaviors, whereas MA-AIRL converged exclusively to competitive behavior (Table 8).

Table 1: Policy imitation performance measured as mean true return on discrete-action tasks

Method	Simple Spread	Simple Push cross-play		Gems
	Shared return ↑	Learned adversary ↑	Learned good ↑	Agent-mean return ↑
Expert reference	-21.592	0.722	-7.228	3.581
MA-AIRL	-22.135 ± 0.071	-0.041 ± 0.005	-7.885 ± 0.081	0.516 ± 0.004
MAMQL	-28.098 ± 0.415	0.468 ± 0.003	-7.443 ± 0.008	2.587 ± 0.542
MA-UITD	$\mathbf{-21.848 \pm 0.033}$	$\mathbf{0.586 \pm 0.021}$	$\mathbf{-7.319 \pm 0.022}$	$\mathbf{2.953 \pm 0.099}$

Values are reported as mean ± sample standard deviation across three training seeds, each evaluated over 500 stochastic-policy episodes.

Reward recovery. We evaluate reward recovery on held-out transitions using Pearson's correlation coefficient (PCC), which measures linear agreement, and Spearman's rank correlation coefficient (SCC), which measures agreement in reward ordering. For each agent, we compare the ground-truth immediate reward with the reward predicted by the learned reward function (Table 7). Figure 1 shows that MA-UITD outperforms the baselines on both measures across all three simulated environments.

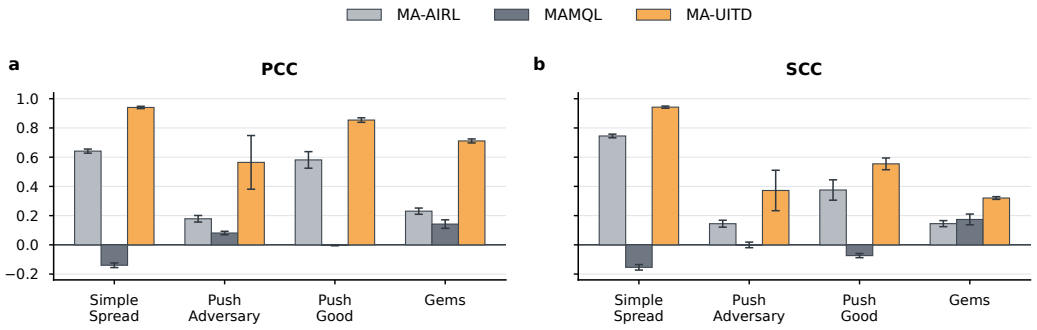


Figure 1: **Reward recovery for MA-UITD and baselines.** Each method is independently trained with three random seeds, and correlations are computed on 500 held-out test episodes for each trained model.

Interpretable reward visualization. We next visualize the reward functions recovered by each method. In *Simple Spread* (Fig. 2), MA-UITD captures the task's underlying coverage structure: once one agent is close to a landmark, the other landmark becomes relatively more rewarding for the other agent, encouraging the agents to distribute across landmarks rather than collapse onto the same target. Consistent with the task's symmetric cooperative objective, MA-UITD recovers nearly identical spatial reward structures for both agents. By contrast, MA-AIRL produces highly localized peaks around individual landmarks, whereas MAMQL assigns high rewards to regions inconsistent with the ground-truth structure and recovers markedly different reward functions across agents. Additional visualizations for *Simple Spread* and the other tasks are provided in Appendix C.3.1.

Continuous action spaces. We next evaluated MA-UITD in *continuous-action* versions of all three environments. The recovered policies achieved returns matching the expert reference in *Gems* and approached expert performance in *Simple Spread*, whereas performance in the competitive *Simple Push* task remained role dependent (Table 2). On held-out trajectories, the recovered rewards showed consistent positive agreement with the ground-truth rewards across all three tasks, with the strongest linear agreement observed in *Simple Spread* (agent-mean PCC 0.828 ± 0.033; Table 3). The

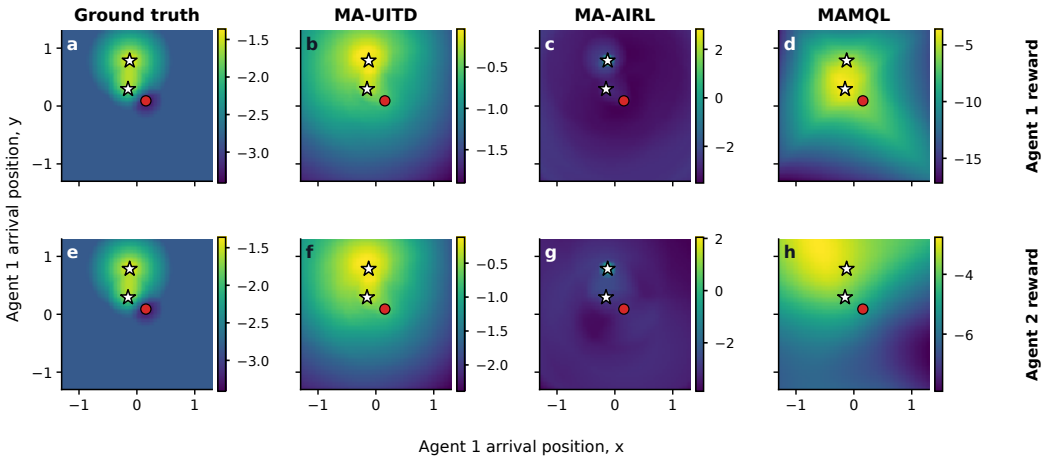


Figure 2: **Visualization of the ground-truth and recovered reward landscapes in *Simple Spread*.** Agent 1's arrival position is varied over the spatial grid while agent 2 (red circle), the landmarks (stars), and arrival velocities are held fixed. The top and bottom rows show the rewards recovered for agents 1 and 2, respectively.

comparatively modest SCC values likely reflect the prevalence of transitions with similar immediate rewards in continuous environments, where small prediction errors can reverse local rankings. We also conducted a qualitative visual analysis by comparing the recovered and ground-truth reward landscapes (Appendix C.3.2). Together, these quantitative and visual results provide evidence that MA-UITD can infer behaviorally compatible reward profiles directly from offline demonstrations with bounded continuous actions.

Table 2: Policy imitation performance of MA-UITD on continuous-control tasks.

Method	Simple Spread	Simple Push cross-play		*Gems*
	Shared return ↑	Learned adversary ↑	Learned good ↑	Agent-mean return ↑
Expert reference	-33.338	-0.063	-6.707	3.337
MA-UITD	-36.783 ± 0.100	-2.286 ± 0.179	-8.611 ± 0.396	3.983 ± 0.026

Values are reported as mean ± sample standard deviation across three training seeds, each evaluated over 500 stochastic-policy episodes.

Table 3: Reward recovery of MA-UITD on continuous-control tasks with held-out demonstrations.

Metric	Simple Spread	Simple Push		*Gems*	
	Agent mean	Adversary	Good	Red	Blue
PCC ↑	0.828 ± 0.033	0.339 ± 0.022	0.717 ± 0.219	0.663 ± 0.129	0.643 ± 0.139
SCC ↑	0.388 ± 0.089	0.322 ± 0.093	0.396 ± 0.035	0.294 ± 0.028	0.305 ± 0.010

Correlations are computed over all transitions in each seed's held-out 500-episode test split and reported as mean ± sample standard deviation across three seeds.

Application to real animal trajectories without trajectory binning. We next evaluated MA-UITD using joint trajectories from pairs of mice performing a social-foraging task (Cheng et al., 2025). In this task, the two animals had to cooperate by navigating to the same reward zone to obtain reward (Fig. 3a). Over training, pairs developed stable asymmetric behavioral roles, with one animal acting as a leader and the other as a follower, potentially reflecting distinct internal reward functions shaped by uncertainty under decentralized, simultaneous movement. Because MA-UITD models a separate reward function for each agent, it can directly represent this form of behavioral asymmetry. Without prior knowledge, however, an animal's internal reward could depend on arbitrary task variables, leaving inverse reward inference severely underconstrained. We therefore restricted

the reward representation to the two animals' positions in the arena and their inter-animal distance, defining reward functions on separate self-position, other-position, and distance components.

Figure 3b,c shows distinct learned reward structures for a representative leader–follower pair. The leader's learned reward depended primarily on its own spatial location, whereas the follower's position-dependent contributions were weaker relative to its proximity-dependent component. This pattern was consistent across mouse pairs: follower rewards increased strongly at short inter-animal distances, while leader rewards showed substantially weaker modulation by distance (Fig. 3d,e). These observations support leader–follower dynamics in which the leader navigates according to spatial goals and the follower is motivated to remain close to the leader.

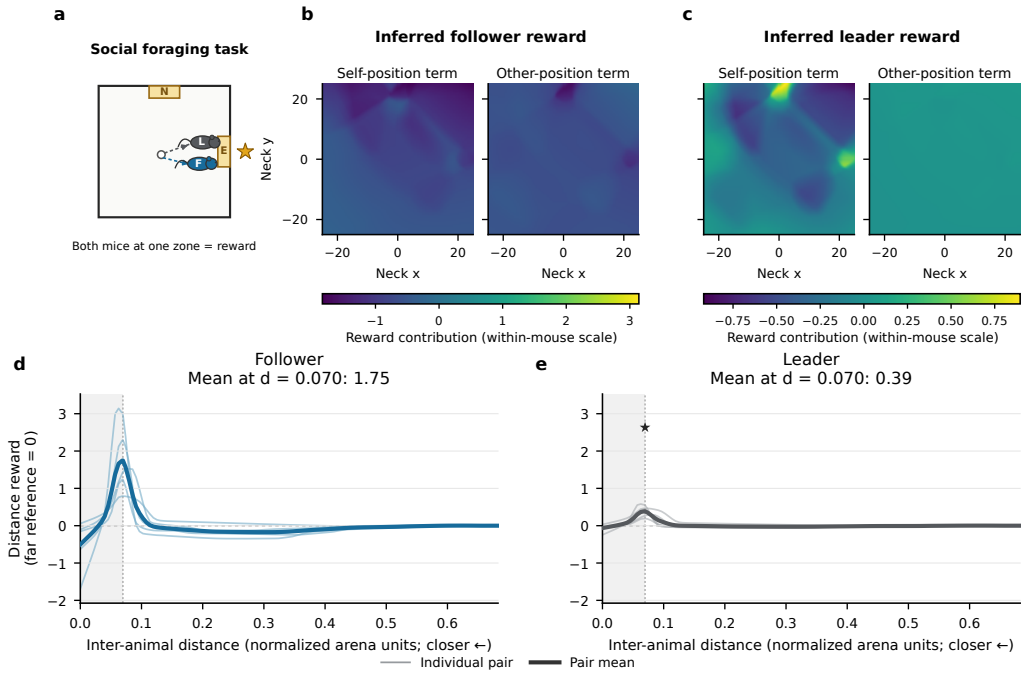


Figure 3: **Learned position and proximity reward components.** **a**, Social-foraging task. **b,c**, Example self- and other-position reward maps, scaled with the distance component using a common within-animal reward range. **d,e**, Inferred distance-dependent rewards for individual pairs (thin) and their mean (thick). Gray shading marks the range below the common 5th-percentile threshold, where data were sparsely sampled. Followers valued proximal partner locations more highly than leaders (two-sided paired t-test at $d = 0.07$, $P = 0.026$; $n = 5$ pairs).

6 Discussion

In summary, we developed MA-UITD, an offline multi-agent inverse reinforcement learning method that uses belief-conditioned successor features to recover explicit agent-specific rewards from fixed demonstrations in continuous-state environments with discrete or bounded continuous actions. The current method conditions reward inference on predictive beliefs estimated from observed behavior; an important direction for future work is to infer each agent's internal beliefs jointly with its reward function and to extend the framework to interactions involving larger numbers of agents. One limitation is that demonstrations provide little evidence about rewards outside the support of the observed states. Reward estimates in these regions therefore depend largely on regularization and the inductive biases of the chosen function approximator, and should be interpreted cautiously until supported by additional data.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language polishing, literature retrieval and discovery (e.g., identifying related work), and coding. It is important to note that generative AI tools were not involved in the ideation, research methodology, or experimental design. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Zoe Ashwood, Aditi Jha, and Jonathan W Pillow. Dynamic inverse reinforcement learning for characterizing animal behavior. *Advances in Neural Information Processing Systems*, 35:29663–29676, 2022.

Chris L. Baker, Julian Jara-Ettinger, Rebecca Saxe, and Joshua B. Tenenbaum. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1(4):0064, March 2017. ISSN 2397-3374. doi: 10.1038/s41562-017-0064. URL https://www.nature.com/articles/s41562-017-0064.

Oscar Bartra, Joseph T. McGuire, and Joseph W. Kable. The valuation system: A coordinate-based meta-analysis of BOLD fMRI experiments examining neural correlates of subjective value. *NeuroImage*, 76:412–427, 2013. doi: 10.1016/j.neuroimage.2013.02.063.

Hannah M. Bayer and Paul W. Glimcher. Midbrain dopamine neurons encode a quantitative reward prediction error signal. *Neuron*, 47(1):129–141, 2005. doi: 10.1016/j.neuron.2005.05.020.

Matteo Bettini, Amanda Prorok, and Vincent Moens. Benchmarl: Benchmarking multi-agent reinforcement learning. *Journal of Machine Learning Research*, 25(217):1–10, 2024. URL http://jmlr.org/papers/v25/23-1612.html.

Ethan S. Bromberg-Martin, Yang-Yang Feng, Takaya Ogasawara, J. Kael White, Kaining Zhang, and Ilya E. Monosov. A neural mechanism for conserved value computations integrating information and rewards. *Nature Neuroscience*, 27(1):159–175, January 2024. ISSN 1546-1726. doi: 10.1038/s41593-023-01511-4. URL https://doi.org/10.1038/s41593-023-01511-4.

Alex J. Chan and Mihaela van der Schaar. Scalable Bayesian Inverse Reinforcement Learning, March 2021. URL http://arxiv.org/abs/2102.06483. arXiv:2102.06483 [cs.LG].

Yusi Chen, Yuan Cheng, Myungji Kwak, Angela Radulescu, and Herbert Zheng Wu. Unveiling value functions in social cognition with multi-agent inverse reinforcement learning. *bioRxiv*, 2026. doi: 10.1101/2024.10.09.617461. URL https://www.biorxiv.org/content/early/2026/04/08/2024.10.09.617461.

Yuan Cheng, Yusi Chen, Myungji Kwak, Ross P. Kempner, Rudramani Singha, Jared Winslow, Runqi Liu, Umais Khan, Tessa Spangler, Alvi Khan, Talmo Pereira, Matthew Whiteway, Evan S. Schaffer, Nuttida Rungratsameetaweemana, Nan Yang, and Herbert Zheng Wu. Asymmetric social representations in the prefrontal cortex for cooperative behavior. *bioRxiv*, 2025. doi: 10.1101/2025.08.27.672249. URL https://www.biorxiv.org/content/early/2025/08/31/2025.08.27.672249.full.pdf.

Will Dabney, Zeb Kurth-Nelson, Naoshige Uchida, Clara Kwon Starkweather, Demis Hassabis, Rémi Munos, and Matthew Botvinick. A distributional code for value in dopamine-based reinforcement learning. *Nature*, 577(7792):671–675, 2020. doi: 10.1038/s41586-019-1924-6. URL https://doi.org/10.1038/s41586-019-1924-6.

Angelos Filos, Clare Lyle, Yarin Gal, Sergey Levine, Natasha Jaques, and Gregory Farquhar. PsiPhi-Learning: Reinforcement Learning with Demonstrations using Successor Features and Inverse Temporal Difference Learning. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 3305–3317. PMLR, July 2021. URL https://proceedings.mlr.press/v139/filos21a.html.

Till Freihaut and Giorgia Ramponi. On Feasible Rewards in Multi-Agent Inverse Reinforcement Learning, November 2025. URL http://arxiv.org/abs/2411.15046. arXiv:2411.15046 [cs.LG].

Keisuke Fujii, Kazushi Tsutsui, Yu Teshima, Makoto Itoh, Naoya Takeishi, Nozomi Nishiumi, Ryoya Tanaka, Shunsuke Shigaki, and Yoshinobu Kawahara. Data-driven simulator of multi-animal behavior with unknown dynamics via offline and online reinforcement learning, October 2025. URL http://arxiv.org/abs/2510.10451. arXiv:2510.10451 [cs.LG].

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Nathaniel Haynam, Adam Khoja, Dhruv Kumar, Vivek Myers, and Erdem Bıyık. Multi-agent inverse Q-learning from demonstrations. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15021–15028. IEEE, 2025. doi: 10.1109/ICRA55743.2025.11128370. URL https://doi.org/10.1109/ICRA55743.2025.11128370.

Julian Jara-Ettinger. Theory of mind as inverse reinforcement learning. *Current Opinion in Behavioral Sciences*, 29:105–110, 2019.

Jingyang Ke, Feiyang Wu, Jiyi Wang, Jeffrey Markowitz, and Anqi Wu. Inverse reinforcement learning with switching rewards and history dependency for characterizing animal behaviors. *arXiv preprint arXiv:2501.12633*, 2025. doi: 10.48550/arXiv.2501.12633.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. URL http://arxiv.org/abs/1412.6980. arXiv:1412.6980 [cs.LG].

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, January 2019. URL http://arxiv.org/abs/1711.05101. arXiv:1711.05101 [cs.LG].

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Neural Information Processing Systems (NIPS)*, 2017.

Richard D. McKelvey and Thomas R. Palfrey. Quantal Response Equilibria for Normal Form Games. *Games and Economic Behavior*, 10(1):6–38, July 1995. ISSN 0899-8256. doi: 10.1006/game.1995.1023. URL https://www.sciencedirect.com/science/article/pii/S0899825685710238.

Nir Moneta, Shany Grossman, and Nicolas W. Schuck. Representational spaces in orbitofrontal and ventromedial prefrontal cortex: task states, values, and beyond. *Trends in Neurosciences*, 47(12):1055–1069, December 2024. ISSN 0166-2236. doi: 10.1016/j.tins.2024.10.005. URL https://doi.org/10.1016/j.tins.2024.10.005.

Christian Neumeyer, Frans A. Oliehoek, and Dariu M. Gavrila. General-Sum Multi-Agent Continuous Inverse Optimal Control. *IEEE Robotics and Automation Letters*, 6(2):3429–3436, April 2021. ISSN 2377-3766, 2377-3774. doi: 10.1109/LRA.2021.3060411. URL https://ieeexplore.ieee.org/document/9357891/.

John P. O'Doherty, Alan Hampton, and Hackjin Kim. Model-based fMRI and its application to reward learning and decision making. *Annals of the New York Academy of Sciences*, 1104:35–53, 2007. doi: 10.1196/annals.1390.022.

Camillo Padoa-Schioppa and John A. Assad. Neurons in the orbitofrontal cortex encode economic value. *Nature*, 441(7090):223–226, 2006. doi: 10.1038/nature04676. URL https://doi.org/10.1038/nature04676.

Wolfram Schultz, Peter Dayan, and P. Read Montague. A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599, 1997. doi: 10.1126/science.275.5306.1593. URL https://doi.org/10.1126/science.275.5306.1593.

Ran Tian, Masayoshi Tomizuka, and Liting Sun. Learning Human Rewards by Inferring Their Latent Intelligence Levels in Multi-Agent Games: A Theory-of-Mind Approach with Application to Driving Data, March 2021. URL http://arxiv.org/abs/2103.04289. arXiv:2103.04289 [cs.AI].

Joey Velez-Ginorio, Max H Siegel, Joshua B Tenenbaum, and Julian Jara-Ettinger. Interpreting actions by attributing compositional desires. In *CogSci*, 2017.

Jiyi Wang, Jingyang Ke, Bo Dai, and Anqi Wu. Learning task-agnostic motifs to capture the continuous nature of animal behavior. *arXiv preprint arXiv:2506.15190*, 2025. doi: 10.48550/arXiv.2506.15190. URL https://arxiv.org/abs/2506.15190.

Feiyang Wu, Ye Zhao, and Anqi Wu. Distributional inverse reinforcement learning. *arXiv preprint arXiv:2510.03013*, 2025. doi: 10.48550/arXiv.2510.03013. URL https://arxiv.org/abs/2510.03013.

Lantao Yu, Jiaming Song, and Stefano Ermon. Multi-Agent Adversarial Inverse Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 7194–7201. PMLR, May 2019. URL https://proceedings.mlr.press/v97/yu19e.html.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

A Algorithms

Algorithm 1 Multi-Agent Unilateral Inverse Temporal Difference Learning (MA-UITD)

Input: Reward-free joint demonstrations $\mathcal{D}$; number of agents n; discount γ; negatives K; temperature τ_{NCE}; proposal distributions $\{\nu_i\}_{i=1}^n$; learning rate α; regularization coefficient λ_w.
Output: Shared reward feature parameters $\boldsymbol{\theta}^\Phi$ and agent-specific parameters $\{\boldsymbol{\eta}_i, w_i\}_{i=1}^n$.

Initialize $\boldsymbol{\theta}^\Phi$, $\{\boldsymbol{\eta}_i, w_i\}_{i=1}^n$, and target parameters $\{\tilde{\boldsymbol{\eta}}_i\}_{i=1}^n$
while training budget remains **do**
  Sample a minibatch $\mathcal{B} \sim \mathcal{D}$
  for $i = 1, \dots, n$ **do**
    Sample K unilateral alternatives from $\nu_i(\cdot \mid s)$
    Compute $\mathcal{L}_i^{\text{NCE}}$ using Equation (10)
    $\boldsymbol{\eta}_i \leftarrow \boldsymbol{\eta}_i - \alpha \nabla_{\boldsymbol{\eta}_i} \mathcal{L}_i^{\text{NCE}}$
    $w_i \leftarrow w_i - \alpha \nabla_{w_i} \left(\mathcal{L}_i^{\text{NCE}} + \lambda_w \|w_i\|_1 \right)$
  end for
  Compute $\mathcal{L}^{\text{UITD}} = n^{-1} \sum_{i=1}^n \mathcal{L}_i^{\text{UITD}}$ using Equation (12)
  $\boldsymbol{\theta}^\Phi \leftarrow \boldsymbol{\theta}^\Phi - \alpha \nabla_{\boldsymbol{\theta}^\Phi} \mathcal{L}^{\text{UITD}}$
  for $i = 1, \dots, n$ **do**
    $\boldsymbol{\eta}_i \leftarrow \boldsymbol{\eta}_i - \alpha \nabla_{\boldsymbol{\eta}_i} \mathcal{L}^{\text{UITD}}$
    Update $\tilde{\boldsymbol{\eta}}_i$ from $\boldsymbol{\eta}_i$
  end for
end while
return $\widehat{r}_i(s) = \Phi(s; \boldsymbol{\theta}^\Phi)^\top w_i$ for all i

B Implementation Details

For *Simple Spread*, the reward-feature network takes the scalar agent–landmark and inter-agent distances as input, whereas the BCSF networks take the agent and landmark positions, agent velocities, and each agent's own action. For *Simple Push*, the reward-feature network takes the agent and landmark positions as input, while the BCSF networks additionally receive the agent velocities and each agent's own action. For *Gems*, a shared torso network, MLP$([32, 32])$, encodes the agent positions. The reward-feature head operates on this shared state representation, while each BCSF head additionally receives the corresponding agent's own action. To ensure a fair comparison, all algorithms were provided with the same task-specific state and action information.

B.1 EXPERT DEMONSTRATIONS.

For the discrete-action versions of *Simple Spread* and *Simple Push*, we generate demonstrations using expert policies trained with the multi-agent ACKTR implementation of Yu et al. (2019). For their continuous-action versions and for both versions of the *Gems* game, we use expert policies trained with MASAC (Bettini et al., 2024). The *Gems* datasets contain an equal mixture of fully cooperative and fully competitive expert behavior. Across tasks, the demonstration datasets contain between 500 and 4,000 episodes.

B.2 MA-UITD ON DISCRETE ACTION DATASETS

MA-UITD naturally generalizes to datasets with discrete action spaces by simply replacing NCE-loss in Eq. (10) with a Maximum Likelihood Estimation(MLE) loss

$$\mathcal{L}_i^{\mathrm{MLE}}(\boldsymbol{\eta}_i, w_i) = -\mathbb{E}_{(s,a_i)\sim\mathcal{D}}\left[\log \frac{\exp\big(\bar{\Psi}_i(s,a_i;\boldsymbol{\eta}_i)^\top w_i/\tau_{\mathrm{MLE}}\big)}{\sum_{\tilde{a}_i\in\mathcal{A}_i}\exp\big(\bar{\Psi}_i(s,\tilde{a}_i;\boldsymbol{\eta}_i)^\top w_i/\tau_{\mathrm{MLE}}\big)}\right]. \tag{13}$$

B.3 NOISE DISTRIBUTION

Let $\rho_E(s,\mathbf{a})$ denote the state–joint-action occupancy induced by the expert demonstrations, and let

$$\rho_{E,i}(s,a_i) = \int \rho_E(s,\mathbf{a})\,\mathrm{d}\mathbf{a}_{-i} \tag{14}$$

be its marginal for agent i. We draw a transition $(s',\mathbf{a}')$ uniformly from $\mathcal{D}_{\mathrm{train}}$ and use a_i' as a negative action at the current state s. Equivalently, the noise distribution is the expert's marginal action distribution,

$$\zeta_i(a_i) = \int \rho_{E,i}(s',a_i)\,\mathrm{d}s', \tag{15}$$

approximated empirically using the training demonstrations.

B.4 CONTINUOUS POLICY ROLLOUT

Rollout initial states were generated using a fixed task-specific random seed, independently of training, and were shared across all model seeds. At each state, we approximated each agent's continuous Boltzmann policy using the same candidate-based procedure. Specifically, we generated $K = 256$ low-discrepancy candidate actions from the agent's bounded action domain and evaluated

$$\widehat{Q}_i(s,a_i) = \overline{\Psi}_i(s,a_i)^\top w_i. \tag{16}$$

The executed action was sampled from the categorical approximation

$$\widehat{\pi}_i(a_i \mid s) = \frac{\exp\Big(\widehat{Q}_i(s,a_i)/\tau\Big)}{\sum_{a_i'\in\mathcal{A}_{i,K}}\exp\Big(\widehat{Q}_i(s,a_i')/\tau\Big)}, \qquad \tau = 1, \tag{17}$$

where $\mathcal{A}_{i,K}$ denotes the candidate set. Candidate-generation and policy-sampling seeds were fixed across model seeds, yielding paired evaluation conditions.

B.5 HYPERPARAMETERS

We conducted a comprehensive hyperparameter search for all three algorithms included in the comparison. Hyperparameters used by MA-UITD implementations are listed in the table 4, 5, 6.

Table 4: MA-UITD hyperparameters for discrete action datasets

Hyperparameter	Simple Spread	Simple Push	Gems
Reward-feature approximator Φ	MLP($[128, 128]$)	MLP($[128, 128]$)	MLP($[64, 32]$)
BCSF approximator $\bar{\Psi}_i$	MLP($[256, 256]$)	MLP($[256, 256]$)	MLP($[64, 64]$)
Activation function	SiLU	SiLU	LeakyReLU, slope 0.01
ℓ_1 coefficient	0.05	0.05	0.001
Dropout rate	0	0	0.3
Number of dimensions in Φ	6	128	10
n-step return	4	4	1
Discount factor γ	0.95	0.95	0.99
Polyak coefficient τ	10^{-4}	10^{-4}	10^{-4}
Rationality τ_{MLE}	1	1	1
Optimiser	AdamW(Loshchilov & Hutter, 2019) for $\Phi, \bar{\Psi}_i$ (lr$= 10^{-4}$, wd$= 10^{-5}$); Adam(Kingma & Ba, 2017) for w_i(lr$= 10^{-4}$)	AdamW(Loshchilov & Hutter, 2019) for $\Phi, \bar{\Psi}_i$ (lr$= 10^{-4}$, wd$= 10^{-5}$); Adam(Kingma & Ba, 2017) for w_i(lr$= 10^{-4}$)	AdamW(Loshchilov & Hutter, 2019) for $\Phi, \bar{\Psi}_i$ and Adam(Kingma & Ba, 2017) for w_i; BC lr$= 10^{-4}$, UITD lr$= 5\times10^{-4}$

Table 5: MA-UITD hyperparameters for continuous-action datasets

Hyperparameter	Simple Spread	Simple Push	Gems
Reward-feature approximator Φ	MLP($[128, 128]$)	MLP($[128, 128]$)	MLP($[64, 32]$)
BCSF approximator $\bar{\Psi}_i$	MLP($[256, 256]$)	MLP($[256, 256]$)	MLP($[128, 64]$)
Activation function	SiLU	SiLU	LeakyReLU, slope 0.02
ℓ_1 coefficient	0.05	0.05	0.001
Dropout rate	0	0	0.3
Number of dimensions in Φ	64	128	16
n-step return	4	4	1
Discount factor γ	0.95	0.95	0.99
Polyak coefficient τ	10^{-4}	10^{-4}	10^{-4}
Rationality τ_{NCE}	1	1	1
Optimiser	AdamW(Loshchilov & Hutter, 2019) for $\Phi, \bar{\Psi}_i$ (lr$= 10^{-4}$, wd$= 10^{-5}$); Adam(Kingma & Ba, 2017) for w_i(lr$= 10^{-4}$)	AdamW(Loshchilov & Hutter, 2019) for $\Phi, \bar{\Psi}_i$ (lr$= 10^{-4}$, wd$= 10^{-5}$); Adam(Kingma & Ba, 2017) for w_i(lr$= 10^{-4}$)	AdamW(Loshchilov & Hutter, 2019) for $\Phi, \bar{\Psi}_i$ and Adam(Kingma & Ba, 2017) for w_i; BC lr$= 10^{-4}$, UITD lr$= 5\times10^{-4}$

Table 6: MA-UITD hyperparameters for real animal trajectories

Hyperparameter	
State representation	Physical-order neck positions (x_1, y_1, x_2, y_2)
Action representation	Own observed 2-D neck displacement
Reward-feature approximator Φ	Position MLP($[128, 64]$); distance MLP($[64, 32]$)
BCSF approximator $\bar{\Psi}_i$	MLP($[256, 128]$)
Activation function	LeakyReLU, slope 0.01
ℓ_1 coefficient	0.05
Dropout rate	0.3 for both Φ and $\bar{\Psi}_i$
Number of dimensions in Φ	20 $(8 + 8 + 4)$
n-step return	1
Discount factor γ	0.99
Polyak coefficient τ	10^{-4}
Optimiser	AdamW(Loshchilov & Hutter, 2019) for $\Phi, \bar{\Psi}_i$ (lr$= 10^{-4}$, wd$= 10^{-5}$); Adam(Kingma & Ba, 2017) for w_i(lr$= 10^{-4}$)

B.6 Compute Resources

All experiments were conducted on a workstation equipped with a 12-core AMD Ryzen 9 9900X CPU, 64 GB of system memory, and a single NVIDIA GeForce RTX 5080 GPU with 16 GB of memory. Each training run used one GPU. Experiments were implemented in Python 3.10.20 using PyTorch 2.11.0 and CUDA 12.8.

C Extended Results

C.1 Reward recovery for MA-UITD and baselines

Details for reward recovery performance across MA-UITD and baselines are shown in Table 7

Table 7: Held-out immediate-reward recovery.

Method	Simple Spread	Simple Push		Gems
	PCC ↑ SCC ↑	Adversary PCC ↑ Adversary SCC↑	Good PCC ↑ Good SCC↑	PCC ↑ SCC ↑
	0.641 ± 0.014	0.178 ± 0.022	0.581 ± 0.057	0.230 ± 0.021
MA-AIRL	0.745 ± 0.013	0.145 ± 0.024	0.375 ± 0.070	0.145 ± 0.021
	-0.140 ± 0.016	0.081 ± 0.012	-0.003 ± 0.003	0.142 ± 0.029
MAMQL	-0.154 ± 0.019	-0.000 ± 0.019	-0.074 ± 0.014	0.174 ± 0.037
	$\mathbf{0.940 \pm 0.008}$	$\mathbf{0.564 \pm 0.184}$	$\mathbf{0.854 \pm 0.016}$	$\mathbf{0.711 \pm 0.014}$
MA-UITD	$\mathbf{0.942 \pm 0.007}$	$\mathbf{0.372 \pm 0.138}$	$\mathbf{0.554 \pm 0.040}$	$\mathbf{0.321 \pm 0.009}$

Entries are mean ± sample standard deviation across three training seeds on the fixed 500-episode test split; the upper and lower values in each cell are PCC and SCC, respectively.

C.2 Policy imitation for the *Gems*

Details for policy imitation in *Gems* are shown in Table 8

Table 8: Policy imitation for the *Gems*

Method	Purple gems (%)	Red/Blue gems (%)
Ground Truth	50	50
MA-AIRL	0.0 ± 0.0	100.0 ± 0.0
MAMQL	50.9 ± 10.7	49.1 ± 10.7
MA-UITD	44.8 ± 1.4	55.2 ± 1.4

Percentage of gems collected. Values are reported as the mean ± sample standard deviation across three training seeds, each evaluated over 500 stochastic-policy episodes.

C.3 Reward visualization

In this section, we provide a comprehensive visual analysis of the reward functions recovered from both discrete- and continuous-action datasets(Fig. 4, 5, 6, 7).

C.3.1 RECOVERED REWARD IN DISCRETE-ACTION ENVIRONMENTS

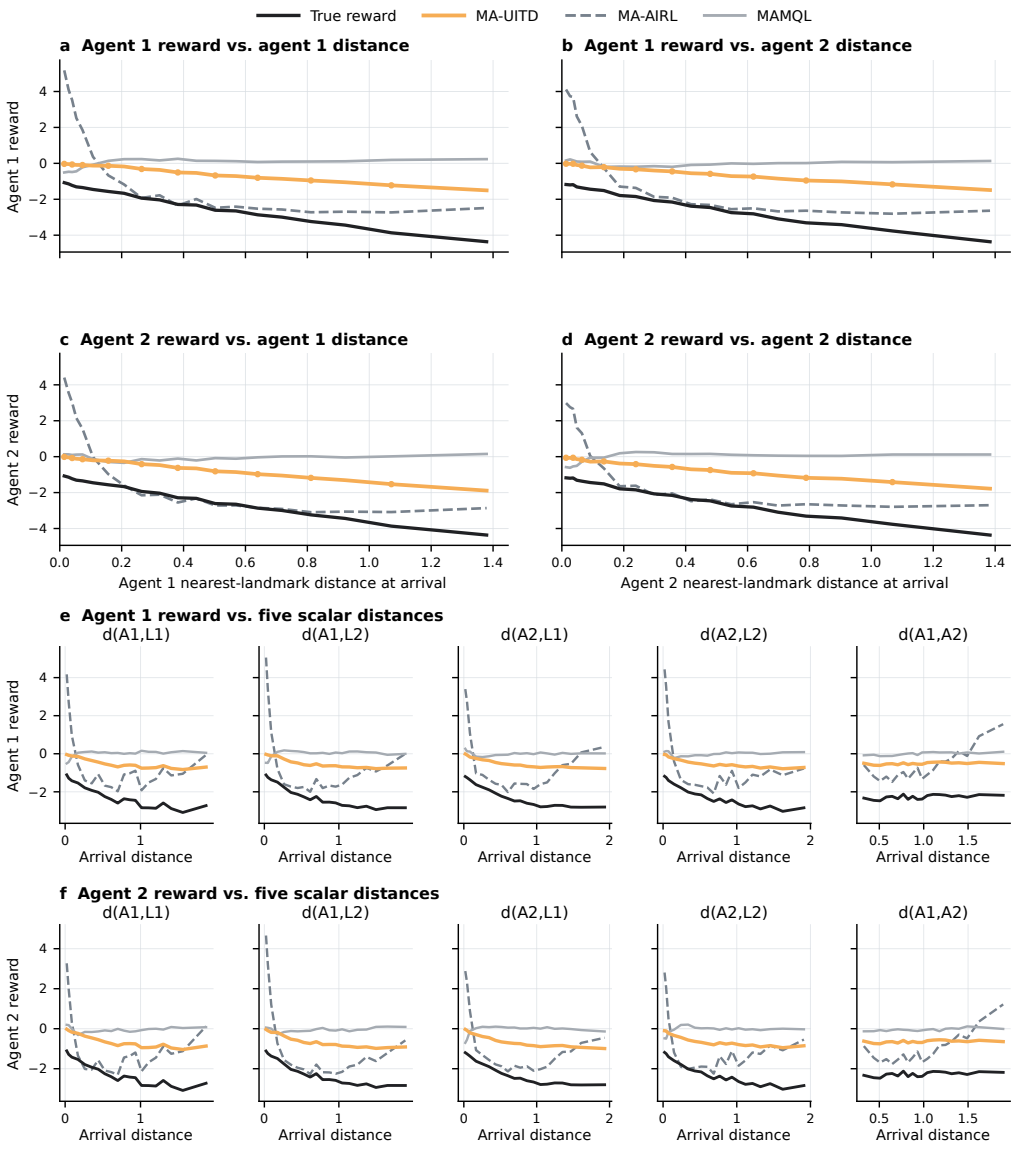


Figure 4: **Reward recovery in discrete *Simple Spread*. a,b**, Agent 1 reward versus each agent's nearest-landmark distance; **c,d**, corresponding Agent 2 results. **e,f**, Agent 1 and Agent 2 rewards versus the five indicated arrival-state distances. Curves show the true and raw MA-UITD, MA-AIRL, and MAMQL rewards, averaged in equal-width bins over 4,836 held-out transitions (500 episodes).

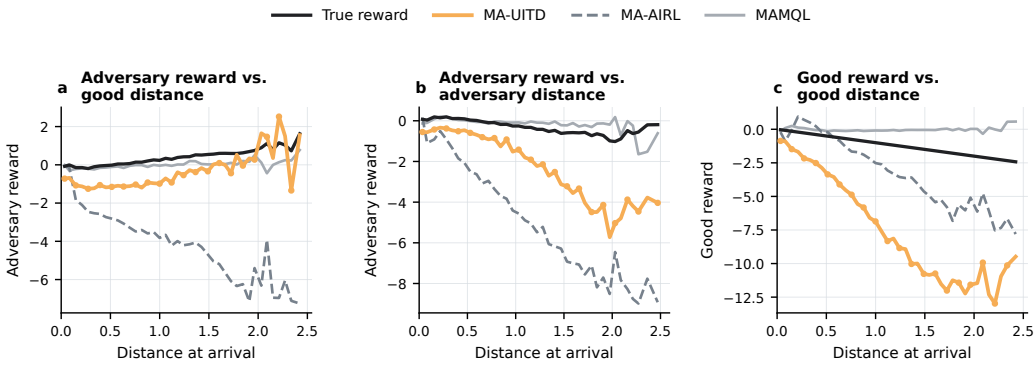


Figure 5: **Reward recovery in discrete *Simple Push*. a,b**, Adversary reward versus the good agent's and its own landmark distances, respectively. **c**, Good-agent reward versus its landmark distance. Curves show the true and raw MA-UITD, MA-AIRL, and MAMQL rewards, averaged in equal-width bins over 12,500 held-out transitions (500 episodes).

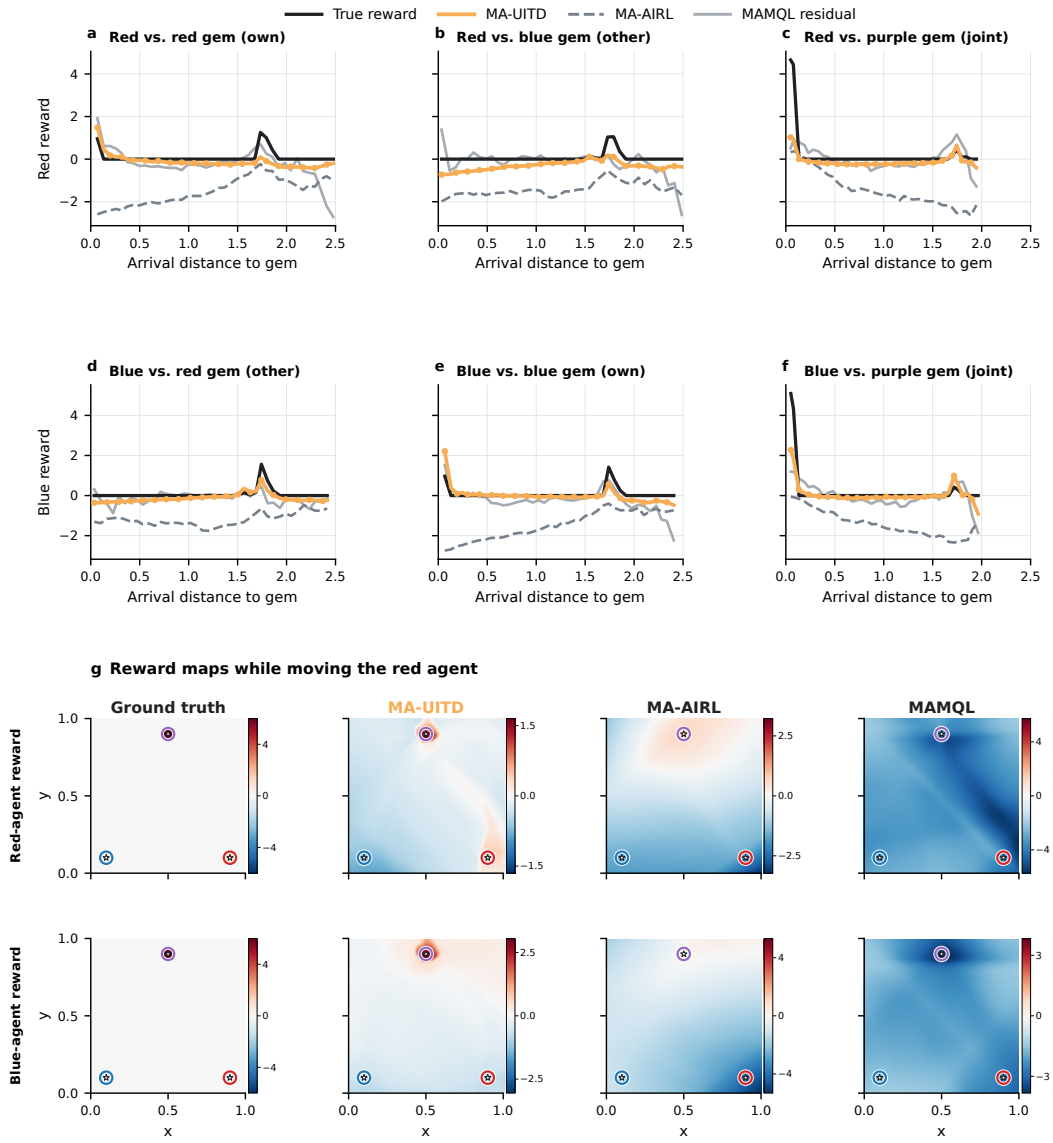


Figure 6: **Reward recovery in discrete *Gems*. a–c**, Red-agent reward versus arrival distance to the red, blue, and purple gems; **d–f**, corresponding blue-agent results. Curves compare ground truth with raw MA-UITD, MA-AIRL, and MAMQL rewards, averaged in equal-width bins over 10,274 held-out transitions (500 episodes). **g**, Spatial reward maps obtained by moving the red agent while fixing the blue agent at the purple gem. Rows show red- and blue-agent rewards; columns show ground truth and the three recovered rewards. Circles mark gem locations and collection radii. Each map uses an independent symmetric scale centered at zero (blue, negative; red, positive).

C.3.2 RECOVERED REWARD IN CONTINUOUS-ACTION ENVIRONMENTS

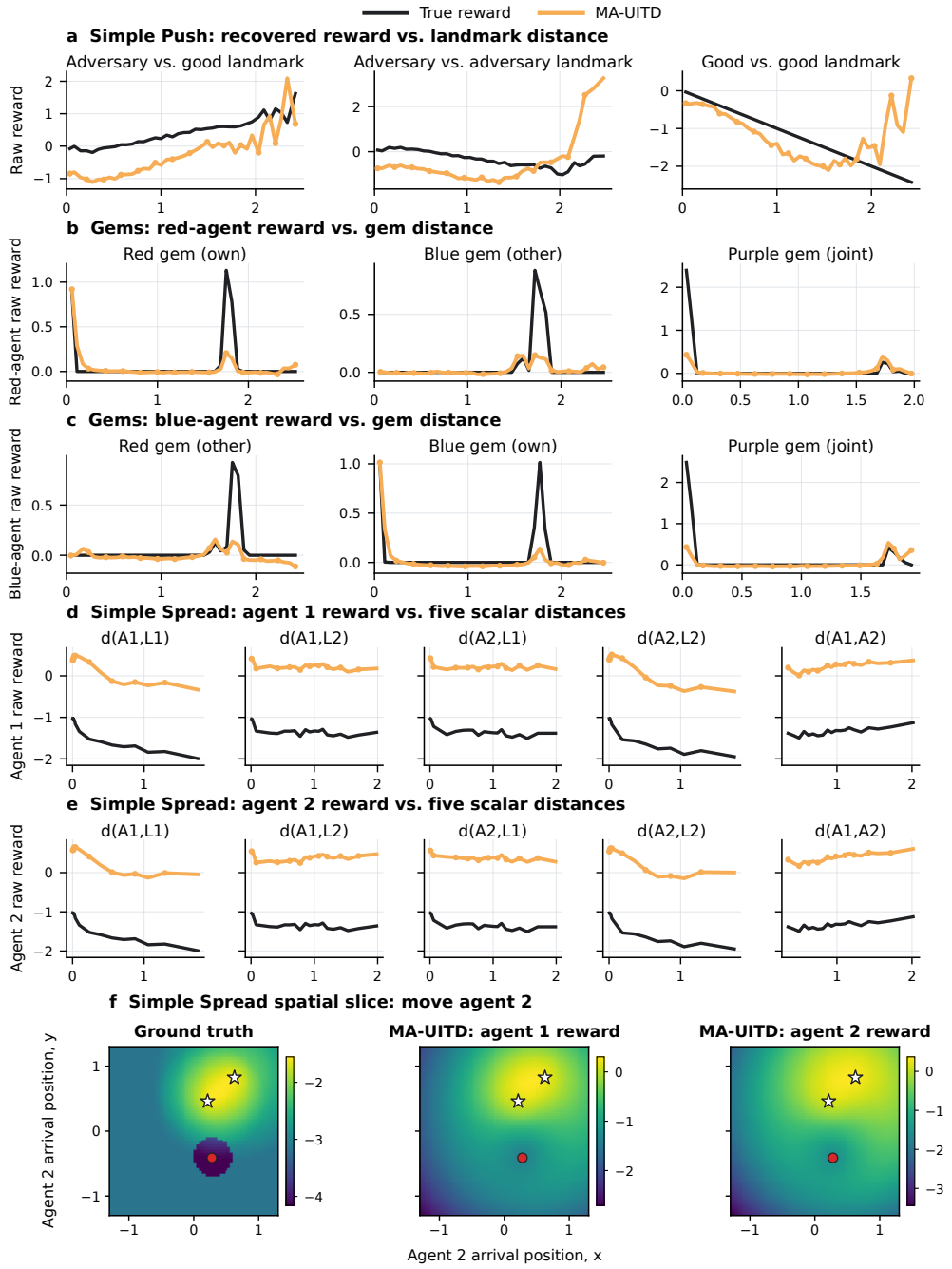


Figure 7: **Reward recovery across continuous-action environments. a**, Reward–distance relationships in *Simple Push*. **b,c**, Red- and blue-agent relationships in *Gems*. **d,e**, Agent 1 and Agent 2 rewards versus five arrival-state distances in *Simple Spread*. **f**, Spatial rewards obtained by moving Agent 2 while fixing Agent 1 and the landmarks. Black and orange denote true and raw MA-UITD rewards. Curves are binned means from 500 held-out episodes per task; Spatial maps use independent raw color scales.