Molten Pot: Evaluations & Datasets for Social Offline Reinforcement Learning

Claude Formanek*
AI Safety South Africa &
University of Cape Town
frmjua001@myuct.ac.za

Marcel Hedman *
Department of Statistics
University of Oxford
marcel.hedman@jesus.ox.ac.uk

Kale-ab Abebe Tessera
School of Informatics
University of Edinburgh

Chris Holmes
Department of Statistics
University of Oxford

Jonathan P. Shock
Department of Mathematics
University of Cape Town

Abstract

Many of the settings where reinforcement learning (RL) could matter most are both social and data-limited: agents must act in the presence of other decision-makers, yet cannot rely on online interaction to learn how to do so. Current evaluations do not target data-constrained social reasoning. Standard offline RL benchmarks treat the environment as non-social, while multi-agent benchmarks focus exclusively on fully cooperative settings. Thus the challenge of reasoning about partner identity and motivations under mixed incentives, and generalising across social structures from offline data alone, remains untested. We introduce Molten Pot, an evaluation protocol, datasets and benchmark for offline mixed-motive social RL built on Melting Pot substrates. The protocol spans five substrates, 47 social scenarios, approximately one terabyte of trajectory data, and defines three complementary evaluation settings that each probe a different aspect of social robustness. Setting 1 tests offline RL in multi-agent, mixed-motive settings with fixed background populations. Setting 2 pools datasets across every scenario of a substrate, requiring the learnt policy to handle varying partner behaviour without explicit context. Setting 3 evaluates zero-shot social generalisation through disjoint train/test splits that isolate specific social shifts. Finally, we benchmark four representative offline RL algorithms on our evaluation protocol and datasets, finding clear limitations in current methods' ability to learn robust social strategies from offline data alone. Molten Pot establishes offline social evaluation as a distinct and necessary target for RL research.

1 Introduction

Humans routinely act appropriately in mixed-motive social situations they have never encountered before [4, 38]. A driver merging into unfamiliar traffic, a negotiator facing an unknown counterpart, a statistician designing an adaptive trial for a patient population whose adherence patterns must be estimated: each must infer the intentions and likely behaviour of others from limited observation and adapt without the luxury of *online* trial and error [3]. This capacity for social reasoning in novel settings is crucial for trusting agents to act in the real world [9]. Yet in many of the situations where such agents would be deployed, abundant interaction data already exists while further online learning is expensive, dangerous, or impractical. Consider autonomous vehicles trained on recorded

*Equal contribution.

Accepted at ICML 2026 Workshop on *Decision-Making from Offline Datasets to Online Adaptation: Black-Box Optimization to Reinforcement Learning.*

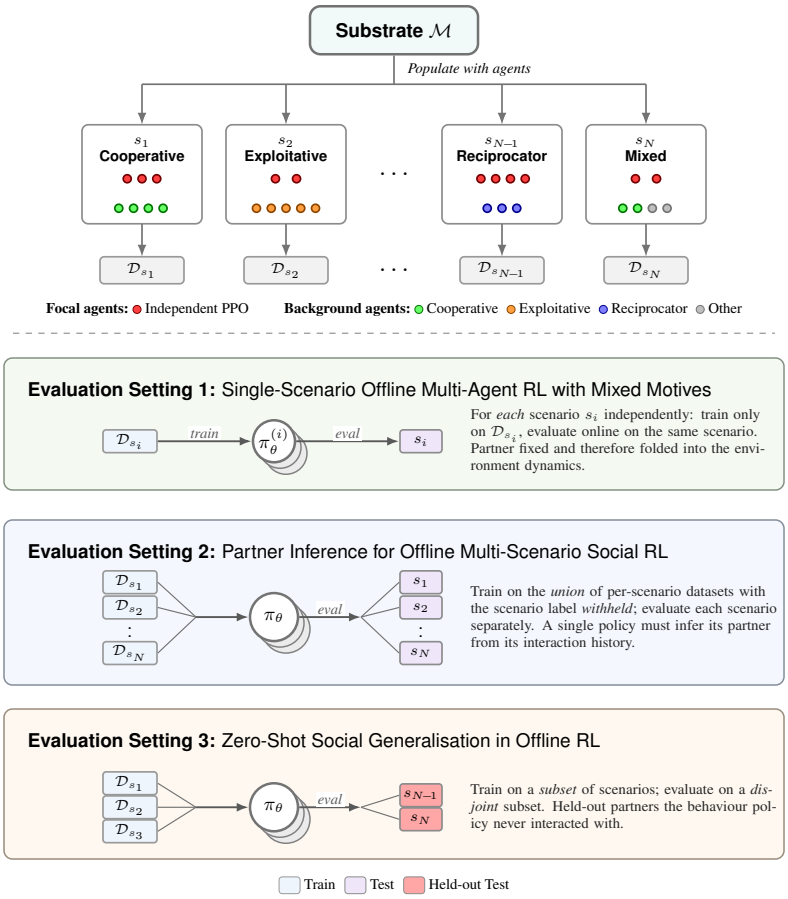


Figure 1: **Molten Pot at a glance.** *Top:* a substrate $\mathcal{M}$ paired with different background populations $\pi_s^{\mathcal{B}}$ defines a family of scenarios $s_i = \langle \mathcal{M}, \pi_{s_i}^{\mathcal{B}} \rangle$. Every scenario hosts multiple *focal* agents, each controlled by its own independent PPO learner; trajectories were logged throughout training so the offline dataset $\mathcal{D}_{s_i}$ mixes skill levels from random initialisation to the converged policy. *Bottom:* the same datasets feed three offline RL evaluation settings that probe distinct facets of social robustness. Evaluation Setting 1 trains and evaluates per scenario independently; Evaluation Setting 2 pools all datasets with the scenario label withheld and tests partner inference; Evaluation Setting 3 trains on a subset of scenarios and evaluates on a disjoint held-out set of scenarios.

driving data alongside human drivers whose behaviour varies and cannot be controlled. Alternatively, consider designing adaptive or platform trials from a corpus of past trials run by different sponsors and sites, whose enrolment, dosing, and early-stopping practices vary across the dataset and cannot be intervened on after the fact [16]. In each case the data is plentiful, the environment is social, and the agent has no opportunity to explore online [16, 26]. We must evaluate social reasoning under the data constraints that deployment frequently imposes. Current evaluation infrastructure does not test this capability.

Offline RL, as benchmarked by D4RL [13], RL Unplugged [18], and their successors, have focused almost exclusively on single-agent control problems, in which the environment dynamics are stationary and the training distribution varies only through the quality of the behaviour policy used to collect data. The evaluation targets these benchmarks stress are distributional shift between the behaviour policy π^β and the learned policy π^θ, and the conservatism required to avoid over-estimating out-of-distribution actions. What they do *not* evaluate are the social reasoning challenges that arise when the environment contains other adaptive agents. In a social environment, the standard offline RL question of whether we can learn a robust policy from historical logs, becomes whether we can learn a robust policy from logs of interactions between a policy and a *population of co-players with mixed motives*, and does that *policy generalise to unseen players*?

Offline multi-agent RL benchmarks move closer but do not reach this goal. OG-MARL and its extensions [10, 12] target offline Multi-Agent RL (MARL) in fully cooperative environments such as SMAC [35] and the MAMuJoCo [32], where the central challenge is coordination [37] rather than handling heterogeneous partner behaviour under misaligned incentives. Mixed-motive settings, in which agents have overlapping but non-identical incentives and must reason about whether and when to cooperate, are largely absent. Melting Pot [24] provides exactly the substrates needed to probe mixed-motive social reasoning, but was designed for online evaluation: policies are trained with self-play or population-based methods and then scored against held-out background populations. The offline counterpart — learning mixed-motive social policies from fixed datasets and evaluating on these substrates — remains unexplored.

We introduce *Molten Pot*, an evaluation protocol with datasets for offline mixed-motive, social reinforcement learning. It is built on five Melting Pot substrates (Figure 2) that together span several social reasoning challenges including: public-goods provision, agent reciprocity, partner choice, commons governance, and trust formation. Molten Pot provides infrastructure to evaluate offline mixed-motive social reinforcement learning: per-scenario offline datasets totalling approximately one terabyte of trajectory data, and three complementary evaluation settings that probe distinct aspects of social robustness (Figure 1).

Evaluation Setting 1 tests single-scenario offline learning with fixed partner populations, mirroring the standard offline MARL setup [10] but now testing the ability to navigate a mixed-motive social environment. *Evaluation Setting 2* pools datasets across every scenario of a substrate and withholds the scenario label, requiring policies to handle heterogeneous partner behaviour without explicit context, whether by explicit partner inference, by learning a robust generalist strategy, or otherwise. *Evaluation Setting 3* designs train and test splits with held-out scenarios that exhibit specific classes of social shift, including: convention mismatch, reciprocity threshold, punishment capability, deployment role, and latent within-episode partner type. This isolates each type of shift and tests whether methods can generalize to partners exhibiting behaviours the training data never contained. The three settings are not arranged on a difficulty ladder; rather they probe different challenges and a method may succeed on one and fail on another for reasons specific to that setting.

Finally, the capability Molten Pot targets is increasingly relevant beyond the classical RL community. Large language model agents are being deployed into open-ended environments [21] where they interact with other agents (other LLM agents, humans, and legacy automated systems) whose policies are unknown and must be inferred from the example interactions seen during pre-training. Molten Pot provides precisely the infrastructure to ask *"how well do these agents handle unfamiliar co-players given only its fixed pre-training data?"*. We view the benchmark as a precursor to the systematic evaluation of deployed LLM-based agents under similar constraints.

2 Preliminaries

2.1 Partially Observable Markov Games

Following Leibo et al. [24], a *substrate* is a multi-agent environment defined by its map, mechanics, and reward structure, independent of the specific players that populate it. Formally, a substrate is modelled as a partially observable general-sum Markov game [17, 19]

$$\mathcal{M} \;=\; \langle \mathcal{N},\, \mathcal{Z},\, \{\mathcal{A}_i\}_{i\in\mathcal{N}},\, \{\mathcal{O}_i\}_{i\in\mathcal{N}},\, \mathcal{T},\, \{r_i\}_{i\in\mathcal{N}},\, \gamma \rangle,$$

with players $\mathcal{N}$, state space $\mathcal{Z}$, per-player discrete action and observation spaces $\mathcal{A}_i, \mathcal{O}_i$, joint transition kernel $\mathcal{T} : \mathcal{Z} \times \prod_i \mathcal{A}_i \to \Delta(\mathcal{Z})$, per-player reward $r_i : \mathcal{Z} \times \prod_i \mathcal{A}_i \to \mathbb{R}$, and discount $\gamma \in [0, 1)$. Here, observations are egocentric 88×88 RGB frames and rewards are individual; games are mixed-motive, meaning reward functions are neither identical (cooperative) nor opposed (zero-sum) unless explicitly stated.

2.2 Focal and Background Agents

Following Leibo et al. [24], the player set is partitioned into *focal* positions $\mathcal{F} \subset \mathcal{N}$, whose policies are learned and evaluated, and *background* positions $\mathcal{B} = \mathcal{N} \setminus \mathcal{F}$, whose policies are fixed and shipped with the benchmark. Within the benchmarked implementations, focal agents share parameters θ across positions; background agents are scripted or pre-trained co-players that act as a black box from the learner's perspective. Parameter sharing is one choice for tackling focal learning.

(a) Clean Up (b) Coins (c) Coop Mining (d) Commons Harvest (e) Allelopathic Harvest

Figure 2: **The five Molten Pot substrates.** Representative rendered frames from DeepMind's Melting Pot [24] covering a range of mixed-motive game structures: common-pool resource dilemmas, public-goods games, dyadic social dilemmas, partner-choice coordination, and convention adoption. Every substrate ships multiple background-partner scenarios: Clean Up (9), Coins (7), Coop Mining (6), Commons Harvest (12), and Allelopathic Harvest (13), for a total of 47 scenarios. Per-scenario descriptions appear in Table 3.

2.3 Scenarios

A *scenario* is a pair $s = \langle \mathcal{M}_s, \pi_s^{\mathcal{B}} \rangle$ comprising a substrate $\mathcal{M}_s$ and a fixed joint background policy $\pi_s^{\mathcal{B}}$. Molten Pot ships a finite set $\mathcal{S}_{\text{all}}$ of scenarios across five substrates (Tables 2–3); each scenario captures a qualitatively distinct social configuration, such as a focal population *visiting* a resident group of reciprocators or *being visited* by a corrigible rider (one that cooperates only after being sanctioned). Fixing a scenario s absorbs the background agents into the environment dynamics: from the focal perspective, background actions $\mathbf{a}^{\mathcal{B}} \sim \pi_s^{\mathcal{B}}(\cdot \mid h_t^{\mathcal{B}})$ are part of the transition function rather than strategic variables. Writing $\mathcal{M}_s^{\mathcal{F}}(\pi_\theta)$ for the resulting focal-perspective dynamics under scenario s and focal policy π_θ, the focal return is

$$J_s(\pi_\theta) \;=\; \mathbb{E}_{\tau \sim \mathcal{M}_s^{\mathcal{F}}(\pi_\theta)}\left[\sum_{t=0}^{T-1} \gamma^t \, r_t^{\mathcal{F}}\right], \qquad r_t^{\mathcal{F}} \;=\; \frac{1}{|\mathcal{F}|}\sum_{i \in \mathcal{F}} r_{i,t}. \tag{1}$$

2.4 Datasets

For a scenario s, an offline dataset $\mathcal{D}_s$ is a collection of focal-perspective episodes generated by some focal agent policies acting in $\mathcal{M}_s$ alongside the fixed background population $\pi_s^{\mathcal{B}}$. Each episode records, for every focal agent $i \in \mathcal{F}$ and timestep t, the observation $o_{i,t}$, action $a_{i,t}$, and reward $r_{i,t}$. Background actions and the underlying environment state are *not* stored: from the learner's perspective, partner behaviour is observable only through its effect on the focal trajectory.

For every scenario s, the focal agents are trained against the fixed background population $\pi_s^{\mathcal{B}}$ using *independent* PPO [36]: each focal agent runs its own PPO learner and updates on its own individual reward, so from any single focal agent's perspective the learning problem is a partially observable stochastic game [17] in which the background is fixed and the other focal agents are concurrent, non-stationary co-learners. This inherits the canonical difficulties of multi-agent RL; non-stationarity from simultaneous focal updates and credit assignment over shared environmental events.

During each run we log $N = 1000$ trajectories of length $T = 1000$ spaced uniformly across training time rather than only at convergence, so $\mathcal{D}_s$ deliberately mixes focal-skill levels, from near-random early-training behaviour through to the converged policy. To validate the quality of our converged policies, the focal agent returns are compared to the returns reported by Leibo et al. [24] on the same scenarios, confirming that the behaviour policy is a reasonable ceiling against which to normalise offline methods. Following the dataset-transparency recommendation of Formanek et al. [12], the aggregated return distribution per substrate is shown in Figure 3 and the per-scenario breakdown in Appendix B.

3 Partner Inference and Bayes Optimality.

Unlike Evaluation Setting 1, Settings 2 and 3 both require focal agents to act well against varying partner populations that are never identified by label. Setting 2 is designed specifically so that maximising its evaluation metric formally requires partner inference. We now show why.

Because background behaviour is unobserved and varies across scenarios, the dynamics experienced by a focal agent depend on a *latent context*: the background partners themselves. This latent-context structure mirrors the social inference problem faced by humans entering unfamiliar social settings: observed behaviour reveals information about the intentions and dispositions of others, which in turn informs action selection [3]. We can formalise this structure. Writing $h_t = (o_0, a_0, r_0, \ldots, o_t)$ for the focal history, the focal-perspective transition under scenario s is obtained by marginalising over background actions:

$$p\big(o_{t+1}, r_t \,\big|\, h_t, a_t;\, s\big) \;=\; \mathbb{E}_{\mathbf{a}^{\mathcal{B}} \sim \pi_s^{\mathcal{B}}(\cdot \,|\, h_t^{\mathcal{B}})}\Big[\, p\big(o_{t+1}, r_t \,\big|\, h_t, a_t, \mathbf{a}^{\mathcal{B}}\big) \,\Big]. \tag{2}$$

When focal agents are deployed against a scenario drawn from a set $\mathcal{S}$ with prior $\Pr(s)$, the Bayes-filter posterior over scenarios given the observed history is

$$b_t(s) \;:=\; \Pr(s \mid h_t) \;\propto\; \Pr(s) \prod_{u=0}^{t-1} p\big(o_{u+1}, r_u \,\big|\, h_u, a_u;\, s\big). \tag{3}$$

As focal agents accumulate observations and rewards, each timestep provides evidence about which partner population it faces, and the belief b_t concentrates on the true scenario. The Bayes-optimal focal policy acts under the belief-weighted predictive dynamics $p(o_{t+1}, r_t \mid h_t, a_t) = \sum_s b_t(s)\, p(o_{t+1}, r_t \mid h_t, a_t; s)$, and the value of the optimal such policy satisfies the Bellman equation

$$V^\star(h_t) \;=\; \max_{a_t}\; \mathbb{E}_{s \sim b_t}\Big[\, \mathbb{E}_{r_t,\, o_{t+1} \sim p(\cdot \,|\, h_t, a_t;\, s)}\big[\, r_t + \gamma\, V^\star(h_{t+1}) \,\big] \,\Big]. \tag{4}$$

This is the optimal expected return for the scenario distribution $\mathcal{S}$ under prior $\Pr(s)$: because $V^\star$ solves the maximisation over all history-conditioned policies given exact knowledge of each scenario's generative model and exact Bayesian inference, no other history-conditioned policy achieves higher expected return averaged over scenario draws from the same prior. On any individual realised scenario a policy better calibrated to that specific partner population may achieve higher return, but only by sacrificing performance on others; $V^\star$ is the optimum over the distribution as a whole, and serves as an upper bound on the expected performance of any learned policy, where both the generative model and the posterior must additionally be approximated from finite offline data. The tension between per-scenario specialisation and distributional robustness is precisely what the evaluation protocol is designed to expose.

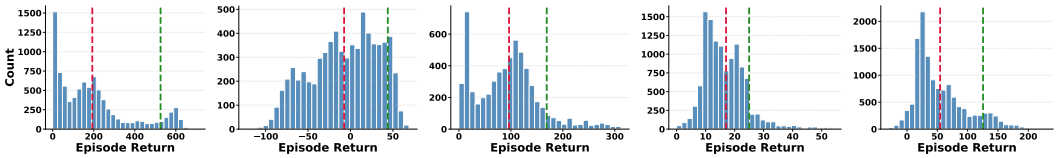
This Bayesian formulation characterises what must be solved, not how. An agent that matches the Bayes-optimal policy through explicit posterior tracking and an agent that arrives at the same value through a learned generalist strategy are equally optimal; in the latter case, the requisite social reasoning is occurring implicitly. Maximising expected return averaged over a scenario distribution, where partner identity is unobserved and inferable only from within-episode history, therefore formally requires partner inference, whether performed explicitly or implicitly. The evaluation protocol below, in particular Setting 2, operationalises this directly.

4 The Molten Pot Evaluation Protocol

Molten Pot defines three *evaluation settings* that probe different aspects of social robustness in offline RL. Each is a fully specified offline reinforcement-learning problem; what distinguishes them is the support of the training and evaluation scenario distributions. The five substrates, the full scenario catalogue, and the train/test splits of evaluation setting 3 are documented in Appendix A (Tables 2–4).

Formally, we write $\mathcal{D}_{\mathcal{S}} := \bigcup_{s \in \mathcal{S}} \mathcal{D}_s$ for the union of the per-scenario datasets over a set of scenarios $\mathcal{S} \subseteq \mathcal{S}_{\text{all}}$. Each evaluation setting specifies a *training* scenario set $\mathcal{S}^{\text{tr}}$, whose datasets the learner may use, and a *test* scenario set $\mathcal{S}^{\text{te}}$, in which the learned focal policy π_θ is scored. The settings differ in how these two sets relate: identical in Settings 1 and 2, disjoint in Setting 3. A learner is given $\mathcal{D}_{\mathcal{S}^{\text{tr}}}$ and must return π_θ without any environment interaction. In every setting the scenario label s is withheld both from the observation at training time and from the policy at test time: the algorithm only ever sees focal (o, a, r) tuples. Our datasets are released in full on our Hugging Face repository[2].

[2] `https://huggingface.co/datasets/jcformanek/moltenpot`



(a) Clean Up (b) Coins (c) Coop Mining (d) Commons Harvest (e) Allelopathic Harvest

Figure 3: **Per-substrate focal-agent return distributions.** Each panel pools all trajectories logged during the independent-PPO runs of every scenario in the substrate. The spread of episode returns confirms that every substrate's dataset is skill-mixed rather than expert-only. The dashed red line in each panel marks the substrate's mean episode return, and the dashed green line marks the 90th percentile. Across the benchmark we logged $\sim$**1 TB** of focal-agent trajectories. Per-scenario return breakdowns appear in Appendix B.

4.1 Evaluation Setting 1: Single-Scenario Offline Multi-Agent RL with Mixed Motives

Setup. Setting 1 is run independently per scenario: for a given $s \in \mathcal{S}_{\text{all}}$, we set $\mathcal{S}^{\text{tr}} = \mathcal{S}^{\text{te}} = \{s\}$. The algorithm is trained on $\mathcal{D}_s$ and evaluated online in $\mathcal{M}_s$. Performance is the focal return from (1) evaluated in scenario s:

$$J_{\text{I}}(\pi_\theta;\, s) \;=\; J_s(\pi_\theta), \qquad s \in \mathcal{S}_{\text{all}}. \tag{5}$$

What this setting evaluates. Because s is fixed, $\pi_s^{\mathcal{B}}$ is a fixed feature of the environment dynamics and no partner inference is required. Setting 1 therefore mirrors the standard single-task offline POSG from Formanek et al. [10], but in a mixed-motive multi-agent environment instead of a cooperative one. This highlights a gap in existing evaluation: offline RL benchmarks do not test mixed-motive social environments, and multi-agent offline benchmarks are restricted to fully cooperative settings [10, 11]. Setting 1 fills that gap directly, asking whether standard offline methods can learn to act well alongside partners who do not share rewards.

4.2 Evaluation Setting 2: Partner Inference for Offline Multi-Scenario Social RL

Setup. Setting 2 is run independently per substrate: for a given substrate, let $\mathcal{S}_{\text{sub}} \subseteq \mathcal{S}_{\text{all}}$ denote its scenarios. We set $\mathcal{S}^{\text{tr}} = \mathcal{S}^{\text{te}} = \mathcal{S}_{\text{sub}}$ and train a *single* focal policy on the union dataset $\mathcal{D}_{\mathcal{S}_{\text{sub}}}$. Crucially, the scenario label is *not* provided to the learner. The trained policy is then deployed and evaluated in each scenario separately. The headline score is the mixture return

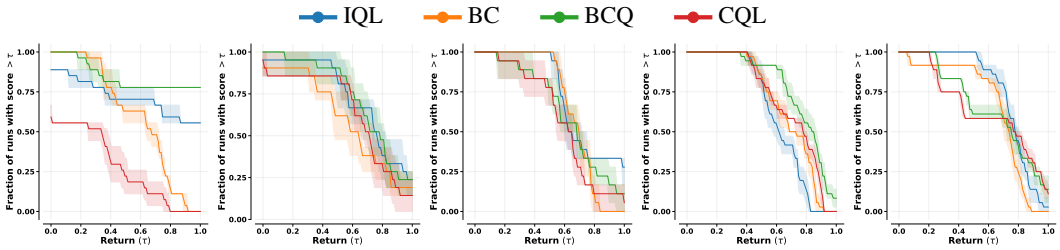
$$J_{\text{II}}(\pi_\theta) \;=\; \mathbb{E}_{s\sim P^{\text{te}}(\mathcal{S}_{\text{sub}})}\big[\, J_s(\pi_\theta)\,\big], \tag{6}$$

where P^{te} is the uniform distribution over $\mathcal{S}_{\text{sub}}$. We also report the per-scenario profile $\{J_s(\pi_\theta)\}_{s\in\mathcal{S}_{\text{sub}}}$ so that scenario specific failures are visible (Fig. 5).

What this setting evaluates. Setting 2 instantiates the evaluation in §3: because evaluation episodes are independent, the policy is deployed fresh in each scenario and must infer the partner population from within-episode history alone. The headline metric J_{II} averages across scenarios under the uniform prior, so its maximiser is the Bayes-optimal partner-inferring policy. A policy that ignores partner identity and plays a single marginal strategy is sub-optimal. The inverse cannot be claimed: poor performance need not stem from missing partner inference alone. High performance also requires generative model specification from finite offline data, and standard offline-RL errors such as extrapolation error, insufficient support, and conservative bias can also drive poor performance. No existing benchmark provides a setting in which partner inference capability can be measured.

4.3 Evaluation Setting 3: Zero-Shot Social Generalisation in Offline RL

Setup. Setting 3 is run independently per substrate. For each substrate we provide disjoint train/test splits $(\mathcal{S}^{\text{tr}}, \mathcal{S}^{\text{te}})$ with $\mathcal{S}^{\text{tr}} \cap \mathcal{S}^{\text{te}} = \emptyset$ (Table 4). The algorithm is trained on $\mathcal{D}_{\mathcal{S}^{\text{tr}}}$, again with scenario labels withheld, and evaluated on the held-out scenarios. Performance is J_{II} as in Setting 2, but evaluated on the held-out scenarios $\mathcal{S}^{\text{te}}$.



(a) Clean Up (b) Coins (c) Coop Mining (d) Commons Harvest (e) Allelopathic Harvest

Figure 4: **Evaluation 1 — per-substrate performance profiles.** For each substrate we plot the rliable performance profile [2, 15]: the fraction of scenarios on which each algorithm scores at least the normalised final-eval return given on the x-axis. Shaded bands are stratified-bootstrap CIs.

Formal view. Setting 3 introduces a distribution shift $P^{\mathrm{tr}}(s) \neq P^{\mathrm{te}}(s)$ with disjoint scenario support: the partner posterior (Eq. 3) is *miscalibrated by construction*, and the Bayes-optimal rule (4) cannot be evaluated directly. The setting is nonetheless not designed to be unsolvable: the observation space, action space, and substrate mechanics are shared across all three settings, so the distributional shift is in the composition of familiar elements rather than in the elements themselves.

What this setting evaluates. Setting 3 introduces the evaluation of zero-shot social generalisation from offline data. Where Setting 2 evaluates partner inference over a known scenario set, Setting 3 asks whether learned social reasoning transfers to partner populations entirely absent from the training distribution. Each train/test split isolates a specific, nameable social shift: convention mismatch, threshold shift in reciprocity, punishment-capability shift, role shift, commitment shift, latent within-episode partner type, feedback-loop shift, and population-scale shift. Table 4 in Appendix A lists the eighteen splits across the five substrates and characterises each shift in detail.

5 Benchmark Results

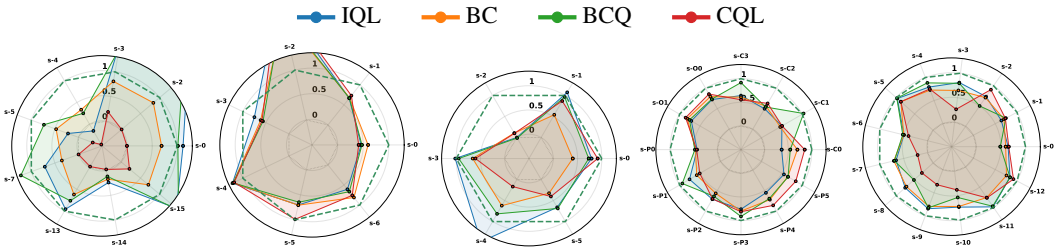
We report results on Evaluation 1, Evaluation 2, and Evaluation 3 using four representative algorithms: behavioural cloning [BC, 34], Batch-Constrained Q-Learning [BCQ, 14], Implicit Q-Learning [IQL, 22], and Conservative Q-Learning [CQL, 23]. Every algorithm's policy and Q-function carries a recurrent GRU [6] layer over the focal agent's interaction history so that partner identity can in principle be inferred from past observations and actions [29]. We use single-agent offline algorithms because existing offline MARL methods target fully cooperative settings via centralised training, which is ill-suited to mixed-motive substrates with distinct per-agent rewards; independent-agent baselines have moreover been shown to be competitive with, and often to outperform, CTDE methods on standardised offline MARL benchmarks [11]. Each algorithm is trained for 20,000 gradient updates with three seeds, and hyperparameters are included in Appendix C. Our code is available on GitHub[3].

Normalisation. For each scenario we report the normalised final eval return $\tilde{J}_s = (R_s - R_{\mathrm{random},s})/(R_{p90,s} - R_{\mathrm{random},s})$ [13], where $R_{\mathrm{random},s}$ is the mean return of a uniform-random focal policy on scenario s and $R_{p90,s}$ is the 90th-percentile episode return in the offline dataset for s (which we use as a proxy for expert policies). Under this normalisation 0 corresponds to a random policy and 1 to the "best-in-dataset" (top-decile) episode. The reference lines on every result figure mark $\tilde{J} = 0$ (grey dotted, random) and $\tilde{J} = 1$ (green dashed, dataset p_{90}).

5.1 Evaluation Setting 1 — Aggregate Performance across Scenarios

Figure 4 reports a *performance profile* per substrate: the fraction of the substrate's scenarios on which each algorithm achieves at least the normalised final-eval return on the x-axis. Curves are computed by MARL-evals utilities [15], with shaded stratified-bootstrap confidence intervals [2]. On most substrates BC produces the most consistent profile — a near-flat plateau followed by a sharp drop at a substrate-specific threshold, indicating that BC scores in a narrow band around roughly

[3] `https://github.com/jcformanek/moltenpot/`



(a) Clean Up (b) Coins (c) Coop Mining (d) Commons Harvest (e) Allelopathic Harvest

Figure 5: **Evaluation 2 — per-substrate scenario profile.** The rays of each spider are that substrate's scenarios (numbered along the perimeter, with Commons Harvest variants abbreviated `C`=closed, `O`=open, `P`=partnership); each algorithm's polygon connects its mean normalised final return on each scenario. The dashed green circle marks the dataset p_{90} ceiling and the dotted grey circle marks the random-policy baseline. Asymmetric polygons reveal scenarios on which an algorithm collapses.

the same return on most scenarios. The offline RL methods (IQL, BCQ, CQL) trace more gradually decreasing curves, reflecting wider per-scenario spread: their curves sit *below* BC's at thresholds inside BC's plateau (scenarios where they underperform BC) but retain non-zero mass to the right of BC's drop-off (scenarios where they substantially exceed BC). This reflects the offline RL algorithms' sensitivity to the varying magnitudes of expected episode return across scenarios: sometimes they fail to learn anything useful, and sometimes they learn a policy that outperforms the dataset's p_{90} episodes by a large margin. Per-scenario breakdowns appear in Appendix D. In Appendix F we also include the change in background agents' returns after the focal agent's policy is deployed. In a few scenarios, especially Coop Mining, the background agents returns increase along with the focal agents, reflecting a *win-win* situation. But in the majority of scenarios, the focal agent's gain is at the expense of the background agents, reflecting the mixed-motive nature of the substrates.

5.2 Evaluation Setting 2 — Per-Substrate Profiles

The substrate-level aggregate hides whether an algorithm is uniformly mediocre or instead does well on a few scenarios and collapses on others. Figure 5 unpacks Setting 2 by substrate, with one spider plot per substrate whose rays are scenarios and whose polygon vertices are each algorithm's mean normalised return. The polygons reveal within-substrate variation that the aggregate erases: on every substrate at least one scenario lies at or below the random-policy reference for every algorithm, and the dent toward the centre falls in broadly the same place across algorithms—consistent with the posterior-inference reading, in which the difficult scenarios are those whose partner type is mismatched against the training mixture rather than the algorithm's choice of conservatism penalty. Per-scenario Setting 1 vs Setting 2 comparisons appear in Appendix D.

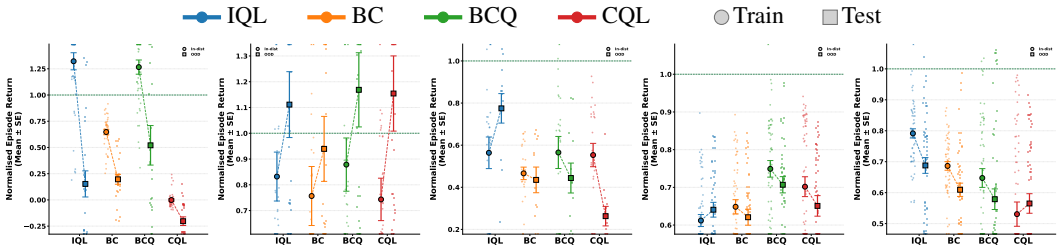
Setting 1 vs Setting 2. Pooling every (substrate, scenario, seed) sample, Table 1 reports RLiable's [2, 15] optimality-gap headline estimator with stratified-bootstrap 95% confidence intervals. Despite Setting 2 training on the union of all per-scenario datasets — substantially more data, drawn from a more diverse mixture than any single Setting 1 run — the optimality gap tends to grow. We tentatively attribute this pattern to the partner-inference pressure that Setting 2 imposes on the learner: withholding the scenario label forces the policy to recover the partner posterior from interaction history alone, and the offline algorithms studied here appear unable to convert the extra data into better generalist social behaviour.

Table 1: **Optimality gap (↓)** pooled across every (substrate, scenario, seed), with rliable 95% CI. All aggregate metrics in Appendix G.

Algo	Setting 1	Setting 2
IQL	0.30 [0.28, 0.33]	0.31 [0.29, 0.32]
BC	0.35 [0.33, 0.36]	0.36 [0.35, 0.37]
BCQ	**0.26 [0.24, 0.27]**	**0.29 [0.27, 0.31]**
CQL	0.43 [0.42, 0.44]	0.49 [0.48, 0.51]

5.3 Evaluation Setting 3 — Train vs. Test Performance.

Figure 6 reports each algorithm's normalised return aggregated across (split, scenario, seed) samples within each substrate, with in-distribution (train, filled circle) and out-of-distribution (held-out test,



(a) Clean Up (b) Coins (c) Coop Mining (d) Commons Harvest (e) Allelopathic Harvest

Figure 6: **Evaluation 3 — per-substrate aggregate normalised return, in-distribution vs out-of-distribution.** For each algorithm: markers are means pooled across every (split, scenario, seed) sample; error bars are SE. Outlier samples beyond the clipped axes are shown as pointing triangles at the edge. Per-split, per-scenario breakdowns in Appendix E.

filled square) evaluations shown side by side. Within every substrate the in-distribution markers sit above their out-of-distribution counterparts: the disjoint-scenario shift costs a measurable fraction of return for every algorithm, and on several substrates pushes out-of-distribution performance back towards the random-policy reference. The gap size varies by algorithm and substrate. Punishment- and role-shifts on Commons Harvest produce the largest drops, consistent with the hypothesis that capability shifts (passive $\rightarrow$ sanctioning counterparts; visitor $\rightarrow$ resident deployment) require composing two transfer problems at once. Coins absorbs its splits most gracefully, where train and test scenarios share a common dyadic structure and differ mainly in partner thresholds. No algorithm closes the in/out gap: scenario-agnostic offline RL recovers within-distribution behaviour but does not produce policies that generalise compositionally to disjoint partner populations. Per-split, per-scenario raw-return breakdowns appear in Appendix E.

6 Related Work

Standard single-agent offline RL benchmarks—D4RL [13] and RL Unplugged [18]—fix per-task behaviour distributions so that conservative algorithms such as BCQ [14], CQL [23], and IQL [22] can be compared on a common axis of distributional shift, but their non-social dynamics make the failure modes Molten Pot targets structurally invisible. The closest multi-agent counterparts are OG-MARL [10] and its revision [12], which inherit the cooperative assumption from SMAC [35] and MAMuJoCo [32] and therefore do not evaluate partner reasoning; offline-MARL algorithmic work has likewise concentrated on cooperative coordination [5, 20, 37], multi-agent conservatism [31], model-based approaches [5, 40], and offline-to-online transfer [27]. The mixed-motive evaluation protocol Molten Pot adopts originates with sequential social dilemmas [25] and Melting Pot [1, 24], whose focal/background split scores generalisation to unseen co-players, and which has since been reimplemented for fast online training in SocialJax [28]; in the mixed-motive online setting, Ndousse et al. [30] further show that social observation can drive agents to learn from one another. Partner handling itself is evaluated in cooperative form by ZSC-Eval [39] and the ad-hoc-teamwork literature, with Mon-Williams et al. [29] showing that recurrent agents form partner models only when the task demands them. Applied mixed-motive MARL has also begun to make its way into human-compatible self-driving [7, 8]. Molten Pot inherits the offline-data discipline of OG-MARL [10, 12] while replacing its cooperative assumption with the focal/background mixed-motive structure of Melting Pot, isolating partner handling rather than coordination as the object of evaluation.

7 Conclusion

We introduced Molten Pot [4], an offline RL evaluation protocol with datasets built from five mixed-motive Melting Pot substrates. Our three Evaluation Settings probe distinct aspects of social robustness: single-scenario fitting (Setting 1), partner inference under withheld scenario labels (Setting 2), and zero-shot generalisation across disjoint train/test partner populations (Setting 3). Four representative algorithms (CQL, BC, BCQ, IQL) leave substantial headroom against the p_{90} dataset ceiling on

[4] Interactive visualization: `https://jcformanek.github.io/moltenpot`

Settings 1 and 2 and incur a clear in-distribution to out-of-distribution drop on Setting 3. Together, the three settings evaluate failure modes that standard offline RL benchmarks overlook. Tackling these settings will require methods that go beyond scenario-agnostic conservatism—partner representations that support extrapolation, posterior-aware conservatism penalties, or memory architectures designed to maintain a calibrated belief over partner type. Molten Pot provides the datasets, splits, and evaluation harness needed to measure progress on each axis, and we hope it serves as a foundation for the next generation of multi-agent offline RL.

Acknowledgments

CF was supported by the Cooperative AI PhD Fellowship from the Cooperative AI Foundation. MH is supported by funding provided by Novo Nordisk and the EPSRC Centre for Doctoral Training in Modern Statistics and Statistical Machine Learning (EP/S023151/1). Computational resources were generously provided by the High Performance Computing (HPC) facility at the University of Cape Town.

References

[1] John P. Agapiou, Alexander Sasha Vezhnevets, Edgar A. Duéñez-Guzmán, Jayd Matyas, Yiran Mao, Peter Sunehag, Raphael Košter, Udari Madhushani, Kavya Kopparapu, Ramona Comanescu, et al. Melting pot 2.0, 2022.

[2] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[3] Chris L. Baker, Rebecca Saxe, and Joshua B. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009. ISSN 0010-0277. DOI: 10.1016/j.cognition.2009.07.005. Reinforcement learning and higher cognition.

[4] 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, 2017.

[5] Paul Barde, Jakob Nicolaus Foerster, Derek Nowrouzezahrai, and Amy Zhang. A model-based solution to the offline multi-agent reinforcement learning coordination problem. *ArXiv*, abs/2305.17198, 2023.

[6] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[7] Daphne Cornelisse and Eugene Vinitsky. Human-compatible driving partners through data-regularized self-play reinforcement learning. *ArXiv*, abs/2403.19648, 2024.

[8] Marco Francis Cusumano-Towner, David Hafner, Alexander Hertzberg, Brody Huval, Aleksei Petrenko, Eugene Vinitsky, Erik Wijmans, Taylor W. Killian, Stuart Bowers, Ozan Sener, Philipp Kraehenbuehl, and Vladlen Koltun. Robust autonomy emerges from self-play. In *Forty-second International Conference on Machine Learning*, 2025.

[9] Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R. McKee, Joel Z. Leibo, K. Larson, and Thore Graepel. Open problems in cooperative ai. *ArXiv*, abs/2012.08630, 2020.

[10] Claude Formanek, Asad Jeewa, Jonathan Shock, and Arnu Pretorius. Off-the-grid marl: Datasets and baselines for offline multi-agent reinforcement learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, pp. 2442–2444, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450394321.

[11] Juan Claude Formanek, Callum Rhys Tilbury, Louise Beyers, Jonathan Phillip Shock, and Arnu Pretorius. Dispelling the mirage of progress in offline MARL through standardised baselines and evaluation. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[12] Juan Claude Formanek, Louise Beyers, Callum Rhys Tilbury, Jonathan Phillip Shock, and Arnu Pretorius. Putting data at the centre of offline multi-agent reinforcement learning. *Journal of Data-centric Machine Learning Research*, 2026.

[13] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

[14] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, 09–15 Jun 2019.

[15] Rihab Gorsane, Omayma Mahjoub, Ruan John de Kock, Roland Dubb, Siddarth Singh, and Arnu Pretorius. Towards a standardised performance evaluation protocol for cooperative marl. *Advances in Neural Information Processing Systems*, 35:5510–5521, 2022.

[16] Omer Gottesman, Fredrik D. Johansson, Joshua Meier, Jack Dent, Donghun Lee, Srivatsan Srinivasan, Linying Zhang, Yi Ding, David Wihl, Xuefeng Peng, Jiayu Yao, Isaac Lage, Christopher Mosch, Li wei H. Lehman, Matthieu Komorowski, A. Aldo Faisal, Leo Anthony Celi, David A. Sontag, and Finale Doshi-Velez. Evaluating reinforcement learning algorithms in observational health settings. *ArXiv*, abs/1805.12298, 2018.

[17] Sven Gronauer and Klaus Diepold. Multi-agent deep reinforcement learning: a survey. *Artif. Intell. Rev.*, 55(2):895–943, February 2022. ISSN 0269-2821. DOI: 10.1007/s10462-021-09996-w.

[18] Caglar Gulcehre, Ziyu Wang, Alexander Novikov, Thomas Paine, Sergio Gómez, Konrad Zolna, Rishabh Agarwal, Josh Merel, Daniel Mankowitz, Cosmin Paduraru, Gabriel Dulac-Arnold, Jerry Li, Mohammad Norouzi, Matt Hoffman, Nicolas Heess, and Nando de Freitas. RL unplugged: A suite of benchmarks for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

[19] Eric A. Hansen, Daniel S. Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, pp. 709–715. AAAI Press, 2004. ISBN 0262511835.

[20] Marcel Hedman, Kale ab Abebe Tessera, Juan Claude Formanek, Anya Sims, Riccardo Zamboni, Trevor McInroe, John Torr, and Elliot Fosong. Coda: Coordination via on-policy diffusion for multi-agent offline reinforcement learning, 2026.

[21] Edward Hughes, Michael D Dennis, Jack Parker-Holder, Feryal Behbahani, Aditi Mavalankar, Yuge Shi, Tom Schaul, and Tim Rocktäschel. Position: Open-endedness is essential for artificial superhuman intelligence. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20597–20616. PMLR, 21–27 Jul 2024.

[22] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *ArXiv*, abs/2110.06169, 2021.

[23] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33: 1179–1191, 2020.

[24] Joel Z. Leibo, Edgar Duénez-Guzmán, Alexander Sasha Vezhnevets, John P. Agapiou, Peter Sunehag, Raphael Koster, Jayd Matyas, Charles Beattie, Igor Mordatch, and Thore Graepel. Scalable evaluation of multi-agent reinforcement learning with melting pot. PMLR, 2021. DOI: 10.48550/arXiv.2107.06857.

[25] Adam Lerer and Alexander Peysakhovich. Maintaining cooperation in complex social dilemmas using deep reinforcement learning, 2018.

[26] Sergey Levine, Aviral Kumar, G. Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *ArXiv*, abs/2005.01643, 2020.

[27] Trevor A. McInroe, Stefano V. Albrecht, and Amos J. Storkey. Planning to go out-of-distribution in offline-to-online reinforcement learning. *ArXiv*, abs/2310.05723, 2023.

[28] Ruoyu Mizuta et al. Socialjax: An evaluation suite for multi-agent reinforcement learning in sequential social dilemmas, 2025.

[29] Ruaridh Mon-Williams, Max Taylor-Davies, Elizabeth Mieczkowski, Natalia Velez, Neil R. Bramley, Yanwei Wang, Thomas L. Griffiths, and Christopher G. Lucas. Partner modelling emerges in recurrent agents (but only when it matters). *arXiv preprint arXiv:2505.17323*, 2025.

[30] Kamal Ndousse, Douglas Eck, Sergey Levine, and Natasha Jaques. Emergent social learning via multi-agent reinforcement learning. In *International Conference on Machine Learning*, 2020.

[31] Ling Pan, Longbo Huang, Tengyu Ma, and Huazhe Xu. Plan better amid conservatism: Offline multi-agent reinforcement learning with actor rectification. *CoRR*, abs/2111.11188, 2021.

[32] Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. Facmac: Factored multi-agent centralised policy gradients. *Advances in neural information processing systems*, 34:12208–12221, 2021.

[33] Julien Perolat, Joel Z. Leibo, Vinicius Zambaldi, Charles Beattie, Karl Tuyls, and Thore Graepel. A multi-agent reinforcement learning model of common-pool resource appropriation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 3646–3655, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[34] Dean A. Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1):88–97, 1991.

[35] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

[36] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[37] Callum Rhys Tilbury, Juan Claude Formanek, Louise Beyers, Jonathan Phillip Shock, and Arnu Pretorius. Coordination failure in cooperative offline MARL. In *ICML 2024 Workshop: Aligning Reinforcement Learning Experimentalists and Theorists*, 2024.

[38] Michael Tomasello, Malinda Carpenter, Josep Call, Tanya Behne, and Henrike Moll. Understanding and sharing intentions: The origins of cultural cognition. *Behavioral and Brain Sciences*, 28(5):675–691, 2005. DOI: 10.1017/S0140525X05000129.

[39] Xihuai Wang et al. ZSC-Eval: An evaluation toolkit and benchmark for multi-agent zero-shot coordination. In *Advances in Neural Information Processing Systems*, 2024.

[40] Daniël Willemsen, Mario Coppola, and Guido C.H.E. de Croon. Mambpo: Sample-efficient multi-robot reinforcement learning using learned world models. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5635–5640, 2021. DOI: 10.1109/IROS51168.2021.9635836.

Appendix contents

- Appendix A: Benchmark reference tables
- Appendix B: Per-scenario return histograms
- Appendix C: Hyperparameters
- Appendix D: Per-scenario Evaluation Setting 1 vs Evaluation Setting 2 eval returns
- Appendix E: Per-scenario Evaluation Setting 3 eval returns by split
- Appendix F: Per-scenario background-agent return change
- Appendix G: Cross-substrate aggregate scores

A Benchmark reference tables

This appendix lists, for reference, the five substrates that compose the Molten Pot benchmark, the full per-scenario catalogue with its social-role badges, and the Evaluation Setting 3 train/test splits.

Table 2: Molten Pot substrates and the primary social challenge each substrate is intended to probe.

Substrate	Players	Social challenge
Clean Up	7	Public-goods maintenance, contribution vs. free-riding, reciprocity, sanctioning, and corrigibility under collective dependence.
Coins [25]	2	Reciprocity and adaptation to partner policy in a mixed-motive dyadic dilemma.
Coop Mining	6	Trust formation, partner choice.
Commons Harvest [33]	7	Tragedy of the commons.
Allelopathic Harvest	4–16	Convention adoption and norm-setting: no dominant individual strategy, so the payoff-optimal planting colour depends on what the rest of the population is doing.

Table 3: Scenario overview for the Molten Pot benchmark. V denotes visitor-mode focal populations and R denotes resident-mode focal populations. PRO prosocial / sustainable counterpart, and ANTI antisocial / unsustainable counterpart. Primary scenario families are: RC reciprocity / conditional cooperation, PC partner choice / partner discrimination, CG commons governance / sustainability, TR trust / institutional commitment, and CA convention adoption. Modifier badges indicate the dominant complication: TH harsher (low) reciprocity threshold / suspiciousness, EN punishment / sanctioning, CO corrigibility, NS non-stationary or alternating partner behavior, LT latent partner type, and ST structured.

Substrate	Sc.	Scenario Type	Description
Clean Up	0	V PRO	Visiting an altruistic population.
	2	V ST	Visiting a turn-taking population that cleans first.
	3	V ST	Visiting a turn-taking population that eats first.
	4	R RC TH	Focals are visited by one reciprocator.
	5	R RC TH	Focals are visited by two suspicious reciprocators.
	7	V RC TH	Focals visit a resident group of suspicious reciprocators.
	13	V RC CO	Focals visit easily corrigible reciprocators.
	14	V RC CO TH	Focals visit reciprocators who are corrigible but difficult to convince.
	15	V RC CO NS	Focals visit easily corrigible reciprocators and also bots who alternate between contributing and free riding.
Coins	0	PC LT	Partner is either a pure cooperator or a pure defector.
	1	RC	Partner is a high-threshold (generous) reciprocator.
	2	RC TH	Partner is a low-threshold (harsh) reciprocator.
	3	RC EN	Partner is a high-threshold (generous) strong reciprocator.
	4	RC EN TH	Partner is a low-threshold (harsh) strong reciprocator.
	5	PC PRO	Partner is a cooperator.
	6	PC ANTI	Partner is a defector.
Coop Mining	0	V TR PRO	Visiting cooperators.
	1	V PC	Visiting residents that extract both ores.
	2	V PC ANTI	Visiting defectors.
	3	R PC PRO	Residents visited by a cooperator.
	4	R PC ANTI	Residents visited by a defector.
	5	V PC LT	Find the cooperator partner.
Commons Harvest: Closed	0	V CG ANTI	Focals visit pacifist bots who harvest unsustainably.
	1	R CG ANTI	Focals are resident and visited by pacifist bots who harvest unsustainably.
	2	V CG PRO EN	Focals visit bots who zap and harvest sustainably if they get a chance.
	3	R CG PRO EN	Focals are resident, and are visited by bots who zap and harvest sustainably if they get a chance.
Commons Harvest: Open	0	R CG ANTI EN	Focals are resident and visited by two bots who zap and harvest unsustainably.

Table 3 (continued)

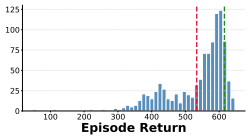
Substrate	Sc.	Scenario Type	Description
	1	R CG ANTI	Focals are resident and visited by two pacifists who harvest unsustainably.
Commons Harvest: Partnership	0	V TR PRO	Meeting good partners. 1 focal agent.
	1	R TR PRO	Focals are resident and visitors are good partners.
	2	V TR PRO EN	Focals visit zappers who harvest sustainably but lack trust.
	3	R TR PRO EN	Focals are resident and visited by zappers who harvest sustainably but lack trust.
	4	V TR ANTI	Focals visit pacifists who do not harvest sustainably.
	5	V TR ANTI EN	Focals visit zappers who do not harvest sustainably.
Allelopathic Harvest	0	V CA ANTI	Visiting a population where planting green berries is the prevailing convention.
	1	V CA PRO	Visiting a population where planting red berries is the prevailing convention.
	2	R CA	Focals are resident and visited by bots who plant either red or green.
	3	CA PRO	Focals like red, visited by convention followers.
	4	CA PRO	Focals like red, visited by mixture of convention followers and bots who like red.
	5	CA ANTI	Focals like red, visited by mixture of convention followers and bots who like green.
	6	CA ANTI	Small-world version of scenario 5 (5 focal, 5 bots).
	7	CA PRO	Small-world pure-follower version of scenario 3 (5 focal, 5 bots).
	8	CA PRO	Minimal 2-focal/2-follower setting;focals like red, meeting convention followers.
	9	R CA	Focal majority with mixed preferences; focals are resident, visited by a couple of convention followers.
	10	R CA	Very small mixed-preference focal group with a single follower bot.
	11	R CA	Large focal majority (8R + 6G) visited by a couple of convention followers.
	12	R CA ANTI	Largest focal majority visited by a couple of convention followers and bots who plant green.

Table 4: Molten Pot Evaluation 3 train–test splits. For each substrate, the split specifies a disjoint $(\mathcal{S}^{\mathrm{tr}}, \mathcal{S}^{\mathrm{te}})$ pair and the social shift it is designed to probe.

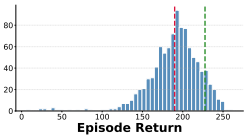
Substrate	Split	Shift	Train	Test	What this split tests
Clean Up	U1	V → V CO	0, 2, 3, 7	13, 14, 15	Train on fixed-behavior bots; test on corrigible bots that respond to being zapped.
	U2	ST CO → TH	0, 2, 3, 15	4, 5, 7	Train on fixed/corrigible bots; test on threshold reciprocators.
	U3	V → R	0, 2, 3, 7, 13	4, 5	Train as visitor minority; test as resident majority.
Coins	C1	PC → RC	0, 5, 6	1, 2, 3, 4	Generalisation from unconditional partner types to conditional reciprocity.
	C2	RC TH → RC	1, 3	2, 4	Transfer from lenient to harsh reciprocity thresholds.
	C3	RC EN → RC	1, 2	3, 4	Transfer across punishment severity.
	C4	PC → PC LT	5, 6	0	Latent partner-type inference under within-episode uncertainty.
Coop Mining	M1	V → R	0, 1, 2, 5	3, 4	Transfer from visitor-mode response to resident-mode competition.
	M2	PC LT → PC	0, 2	5	Generalisation from homogeneous partner populations to heterogeneous populations.
	M3	V PC PC → R	0, 5	3	Generalisation from abundant cooperation opportunities to scarcity and competition for partner access.
Commons Harvest	H1	ANTI EN → ANTI	Closed: 0, 1; Open: 1; Partnership: 4	Open: 0; Partnership: 5	Transfer from passive unsustainable counterparts to enforcement-capable unsustainable counterparts.
	H2	V → R	Closed: 0, 2; Partnership: 0, 2, 4, 5	Closed: 1, 3; Open: 0, 1; Partnership: 1, 3	Transfer from visitor-mode adaptation to resident-mode governance.
	H3	CG → TR	Closed: 0, 1, 2, 3; Open: 0, 1	Partnership: 0, 1, 2, 3	Generalisation from anonymous commons management to dyadic institutional commitment.
	H4	TR PRO PRO → TR EN	Partnership: 0, 1	Partnership: 2, 3	Trust calibration under partner shift from good partners to guarded or adversarial ones.
Allelopathic Harvest	A1	PRO → ANTI	1, 3, 4, 7, 8	0, 5, 6	Generalisation from aligned or shapeable conventions to misaligned or opposing conventions.
	A2	pure followers → mixed visitors	3, 7, 8, 9, 10, 11	4, 5, 6, 12	Robustness of convention-following when some visitors no longer merely follow the prevailing norm but also carry their own colour preference, in both visitor-like and resident-majority settings.
	A3	large → small	3, 4, 5, 11, 12	6, 7, 8, 9, 10	Transfer from large-population convention dynamics to sparse settings where a few agents or even a single plant can determine which convention takes hold.
	A4	V → R	0, 1	2, 9, 10, 11, 12	Transfer from adapting to an already-established visiting convention to sustaining or shaping the convention as a resident focal majority under different visitor mixes.

B Per-scenario return histograms

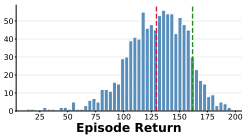
Figures 7–11 show, for every scenario in the benchmark, the distribution of focal-agent episode returns across the 1000 trajectories logged during the scenario's independent-PPO run. Because trajectories are recorded uniformly across training time, each histogram should exhibit a wide support with a left tail from near-random initial rollouts and a right peak near the converged PPO return; uniform sampling across training time deliberately produces a skill-mixed dataset rather than an expert-only one, following the dataset-transparency recommendation of Formanek et al. [12]. Narrow histograms are a warning sign that the scenario's behaviour policy collapsed or that the logging schedule failed to capture the full training arc. In every panel the dashed red line marks the scenario's mean episode return and the dashed green line marks the 90th percentile.



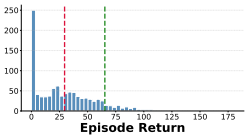
(a) Scenario 0



(b) Scenario 2

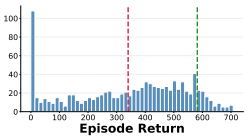


(c) Scenario 3

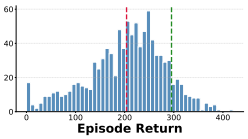


(d) Scenario 4

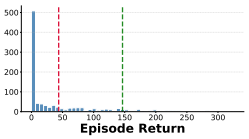
(e) Scenario 5



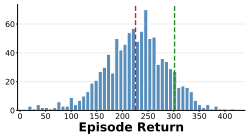
(f) Scenario 7



(g) Scenario 13

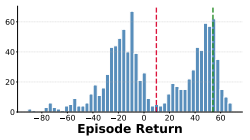


(h) Scenario 14

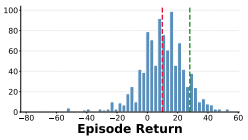


(i) Scenario 15

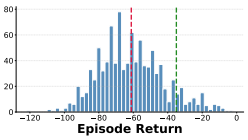
Figure 7: **Clean Up — per-scenario focal-return distributions.** Dashed red: scenario mean. Dashed green: 90th percentile.



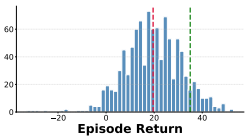
(a) Scenario 0



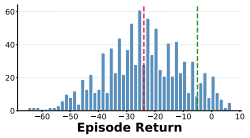
(b) Scenario 1



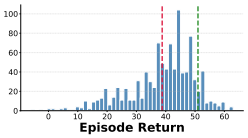
(c) Scenario 2



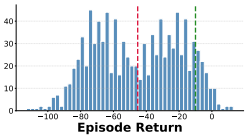
(d) Scenario 3



(e) Scenario 4

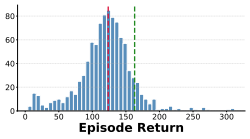


(f) Scenario 5

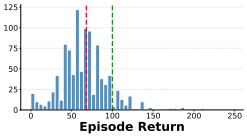


(g) Scenario 6

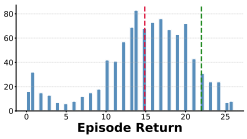
Figure 8: **Coins — per-scenario focal-return distributions.** Dashed red: scenario mean. Dashed green: 90th percentile.



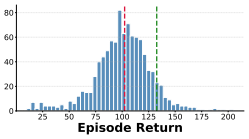
(a) Scenario 0



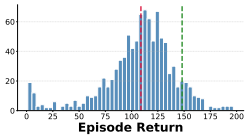
(b) Scenario 1



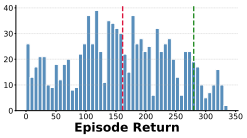
(c) Scenario 2



(d) Scenario 3

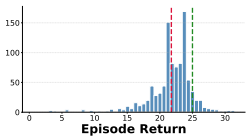


(e) Scenario 4

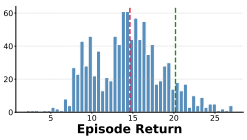


(f) Scenario 5

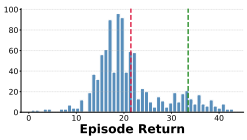
Figure 9: **Coop Mining — per-scenario focal-return distributions.** Dashed red: scenario mean. Dashed green: 90th percentile.



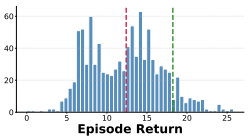
(a) Closed 0



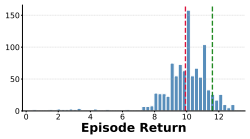
(b) Closed 1



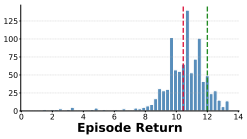
(c) Closed 2



(d) Closed 3

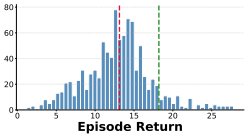


(e) Open 0

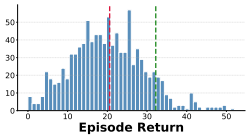


(f) Open 1

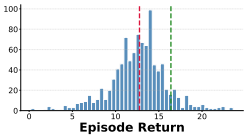
(g) Partnership 0



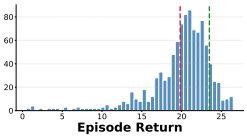
(h) Partnership 1



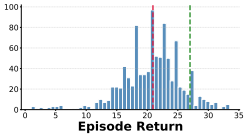
(i) Partnership 2



(j) Partnership 3

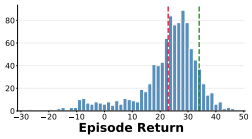


(k) Partnership 4

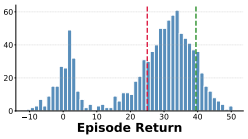


(l) Partnership 5

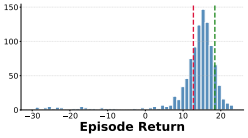
Figure 10: **Commons Harvest — per-scenario focal-return distributions.** Variants are grouped by row: *closed* (top), *open* + *partnership* (middle and bottom). Dashed red: scenario mean. Dashed green: 90th percentile.



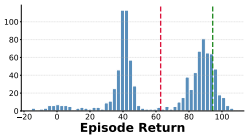
(a) Scenario 0



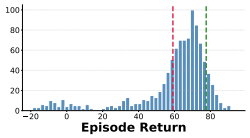
(b) Scenario 1



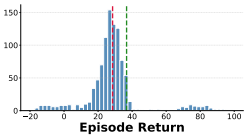
(c) Scenario 2



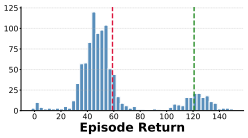
(d) Scenario 3



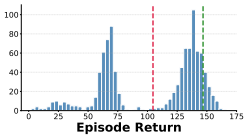
(e) Scenario 4



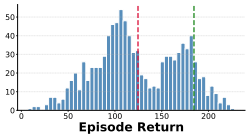
(f) Scenario 5



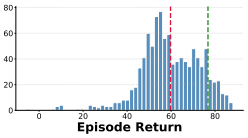
(g) Scenario 6



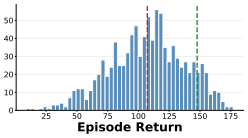
(h) Scenario 7



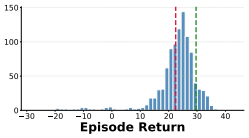
(i) Scenario 8



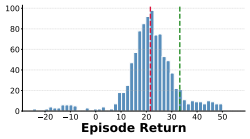
(j) Scenario 9



(k) Scenario 10



(l) Scenario 11



(m) Scenario 12

Figure 11: **Allelopathic Harvest — per-scenario focal-return distributions.** Dashed red: scenario mean. Dashed green: 90th percentile.

C Hyperparameters

Tuning protocol. We purposefully use a modest hyperparameter tuning budget. In particular, hyperparameters are tuned on the *first scenario* of the substrate only, and the resulting values are then reused unchanged on every other scenario in the substrate. It is now well-recognised that for offline RL the online tuning budget should be kept small and fixed, so that comparisons reward algorithms that need less tuning rather than the practitioner who could afford the most online evaluations [11]; in any realistic deployment of an offline-trained policy, online evaluation for hyperparameter selection is exactly the resource we cannot liberally spend.

Shared settings. Every algorithm uses the same focal-agent architecture, optimiser and training schedule (Table 5). The only substrate-dependent hyperparameters are given in Table 6.

Table 5: Shared training and architecture settings, applied across all four offline algorithms and all five substrates.

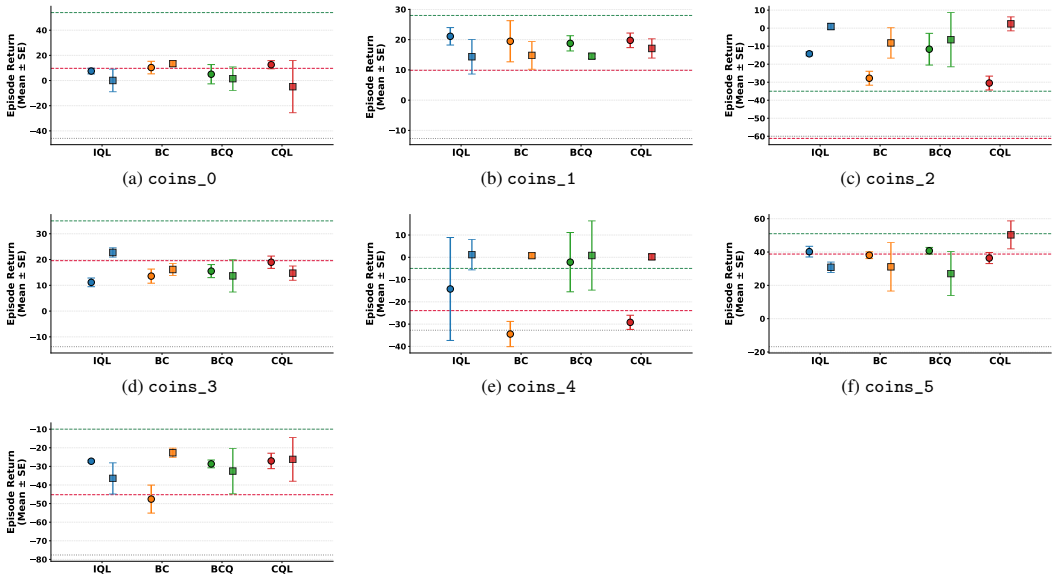
Setting	Value
Convolutional backbone	3-layer CNN (matches MeltingPot 2.4)
Hidden size (FC)	256
Recurrent core (GRU)	256
Optimiser	Adam (default β_1, β_2)
Learning rate	3×10^{-4}
Discount γ	0.99
Soft-target rate τ	0.005
Batch size	32
Sequence length	128
Gradient steps	20,000
Seeds per (substrate, algorithm)	3

Table 6: Per-algorithm, per-substrate hyperparameters. Values were selected by the per-substrate tuning protocol described above (tune on the substrate's first scenario, then re-use for every other scenario in the substrate).

Algorithm	Hyperparameter	Coins	Coop Mining	Clean Up	Commons Harvest	Allelopathic Harvest
IQL	Expectile τ_{IQL}	0.80	0.80	0.70	0.70	0.70
	Advantage temp. β	2.0	4.0	5.0	2.0	3.0
BCQ	Action threshold τ_{BCQ}	0.30	0.30	0.30	0.30	0.40
CQL	CQL coefficient α	3.0	5.0	7.0	3.0	6.0

D Per-scenario Evaluation Setting 1 vs Evaluation Setting 2 eval returns

For each scenario in the benchmark we report the side-by-side Evaluation Setting 1 vs Evaluation Setting 2 eval-return comparison in raw (un-normalised) return units, organised here as one mosaic per substrate. Within each panel the four offline-RL algorithms appear as paired markers: Evaluation Setting 1 as a filled circle (single-scenario training, evaluated on that same scenario) and Evaluation Setting 2 as a filled square (all-scenarios training, evaluated on that scenario). The marker is the mean across the seeds with data and the error bar is the standard error of that mean. Reference lines mark the dataset's mean episode return (red dashed), its p_{90} "best-in-dataset" ceiling (green dashed), and the random-policy baseline (grey dotted). Y-axes are in raw return units, scaled per-panel so within-scenario differences between algorithms remain legible across scenarios with very different return magnitudes.

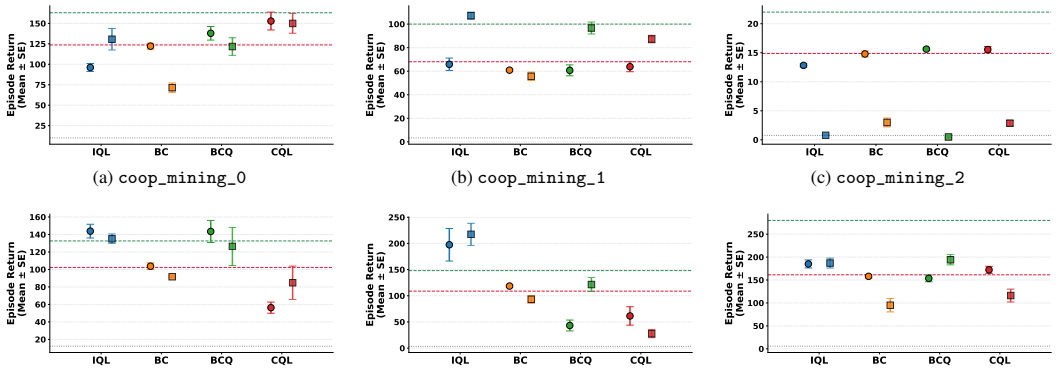


(a) coins_0 (b) coins_1 (c) coins_2

(d) coins_3 (e) coins_4 (f) coins_5

(g) coins_6

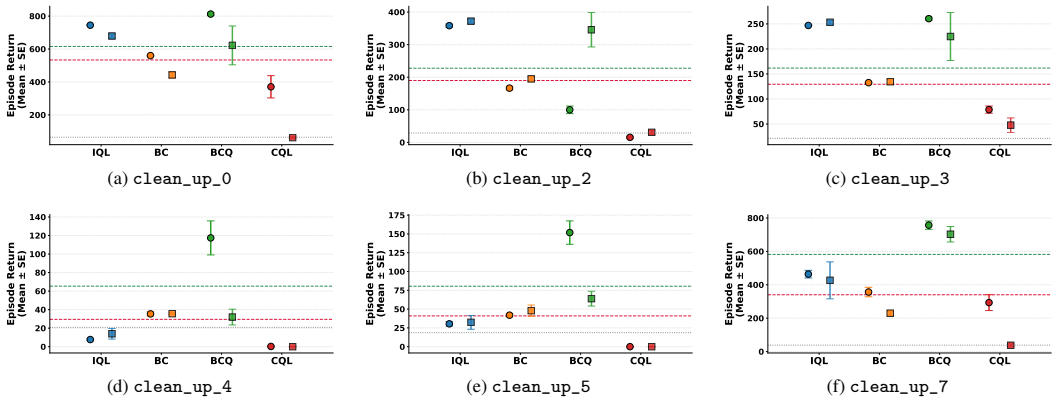
Figure 12: **Per-scenario Evaluation Setting 1 vs Evaluation Setting 2 raw eval returns — Coins.** Each panel shows one scenario in Coins; within each panel the four offline-RL algorithms appear as paired markers — Evaluation Setting 1 (filled circle, single-scenario training) and Evaluation Setting 2 (filled square, all-scenarios training, slightly transparent). Marker = mean across the seeds with data, error bar = SE. Reference lines mark the dataset's mean episode return (red dashed), p_{90} "best-in-dataset" ceiling (green dashed), and the random-policy baseline (grey dotted). Y-axes are in raw return units (no normalisation), scaled per-panel.



(a) coop_mining_0 (b) coop_mining_1 (c) coop_mining_2

(d) coop_mining_3 (e) coop_mining_4 (f) coop_mining_5

Figure 13: **Per-scenario Evaluation Setting 1 vs Evaluation Setting 2 raw eval returns — Coop Mining.** Each panel shows one scenario in Coop Mining; within each panel the four offline-RL algorithms appear as paired markers — Evaluation Setting 1 (filled circle, single-scenario training) and Evaluation Setting 2 (filled square, all-scenarios training, slightly transparent). Marker = mean across the seeds with data, error bar = SE. Reference lines mark the dataset's mean episode return (red dashed), p_{90} "best-in-dataset" ceiling (green dashed), and the random-policy baseline (grey dotted). Y-axes are in raw return units (no normalisation), scaled per-panel.

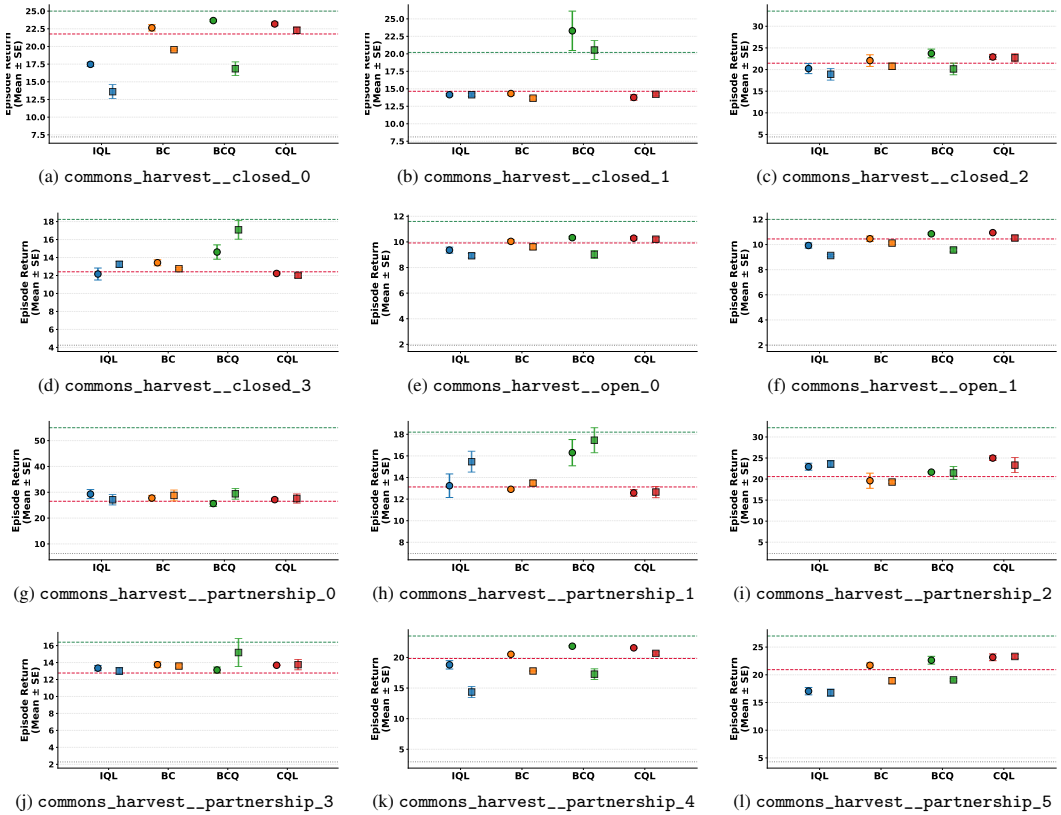


(a) clean_up_0 (b) clean_up_2 (c) clean_up_3

(d) clean_up_4 (e) clean_up_5 (f) clean_up_7

(g) clean_up_13 (h) clean_up_14 (i) clean_up_15

Figure 14: **Per-scenario Evaluation Setting 1 vs Evaluation Setting 2 raw eval returns — Clean Up.** Each panel shows one scenario in Clean Up; within each panel the four offline-RL algorithms appear as paired markers — Evaluation Setting 1 (filled circle, single-scenario training) and Evaluation Setting 2 (filled square, all-scenarios training, slightly transparent). Marker = mean across the seeds with data, error bar = SE. Reference lines mark the dataset's mean episode return (red dashed), p_{90} "best-in-dataset" ceiling (green dashed), and the random-policy baseline (grey dotted). Y-axes are in raw return units (no normalisation), scaled per-panel.



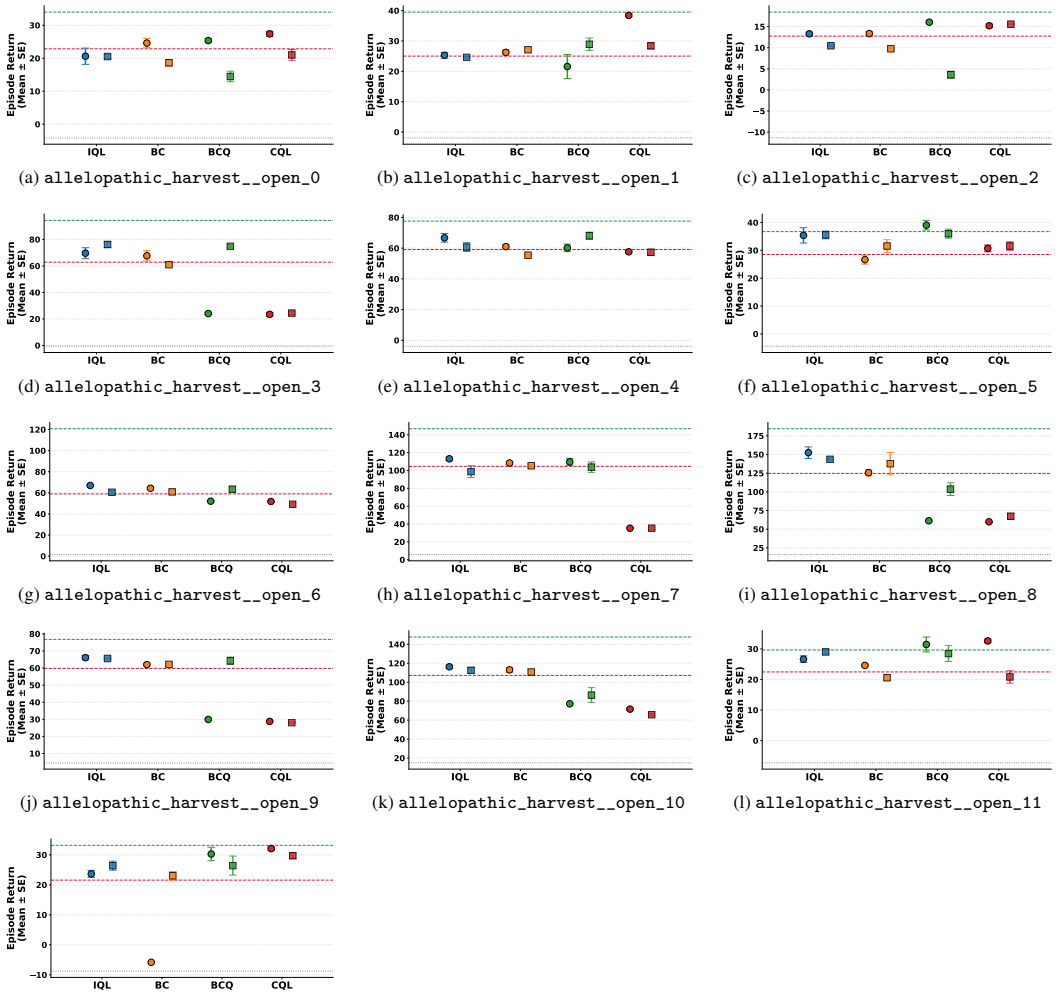
(a) commons_harvest__closed_0 (b) commons_harvest__closed_1 (c) commons_harvest__closed_2

(d) commons_harvest__closed_3 (e) commons_harvest__open_0 (f) commons_harvest__open_1

(g) commons_harvest__partnership_0 (h) commons_harvest__partnership_1 (i) commons_harvest__partnership_2

(j) commons_harvest__partnership_3 (k) commons_harvest__partnership_4 (l) commons_harvest__partnership_5

Figure 15: **Per-scenario Evaluation Setting 1 vs Evaluation Setting 2 raw eval returns — Commons Harvest.** Each panel shows one scenario in Commons Harvest; within each panel the four offline-RL algorithms appear as paired markers — Evaluation Setting 1 (filled circle, single-scenario training) and Evaluation Setting 2 (filled square, all-scenarios training, slightly transparent). Marker = mean across the seeds with data, error bar = SE. Reference lines mark the dataset's mean episode return (red dashed), p_{90} "best-in-dataset" ceiling (green dashed), and the random-policy baseline (grey dotted). Y-axes are in raw return units (no normalisation), scaled per-panel.



(a) allelopathic_harvest__open_0 (b) allelopathic_harvest__open_1 (c) allelopathic_harvest__open_2

(d) allelopathic_harvest__open_3 (e) allelopathic_harvest__open_4 (f) allelopathic_harvest__open_5

(g) allelopathic_harvest__open_6 (h) allelopathic_harvest__open_7 (i) allelopathic_harvest__open_8

(j) allelopathic_harvest__open_9 (k) allelopathic_harvest__open_10 (l) allelopathic_harvest__open_11

(m) allelopathic_harvest__open_12

Figure 16: **Per-scenario Evaluation Setting 1 vs Evaluation Setting 2 raw eval returns — Allelopathic Harvest.** Each panel shows one scenario in Allelopathic Harvest; within each panel the four offline-RL algorithms appear as paired markers — Evaluation Setting 1 (filled circle, single-scenario training) and Evaluation Setting 2 (filled square, all-scenarios training, slightly transparent). Marker = mean across the seeds with data, error bar = SE. Reference lines mark the dataset's mean episode return (red dashed), p_{90} "best-in-dataset" ceiling (green dashed), and the random-policy baseline (grey dotted). Y-axes are in raw return units (no normalisation), scaled per-panel.

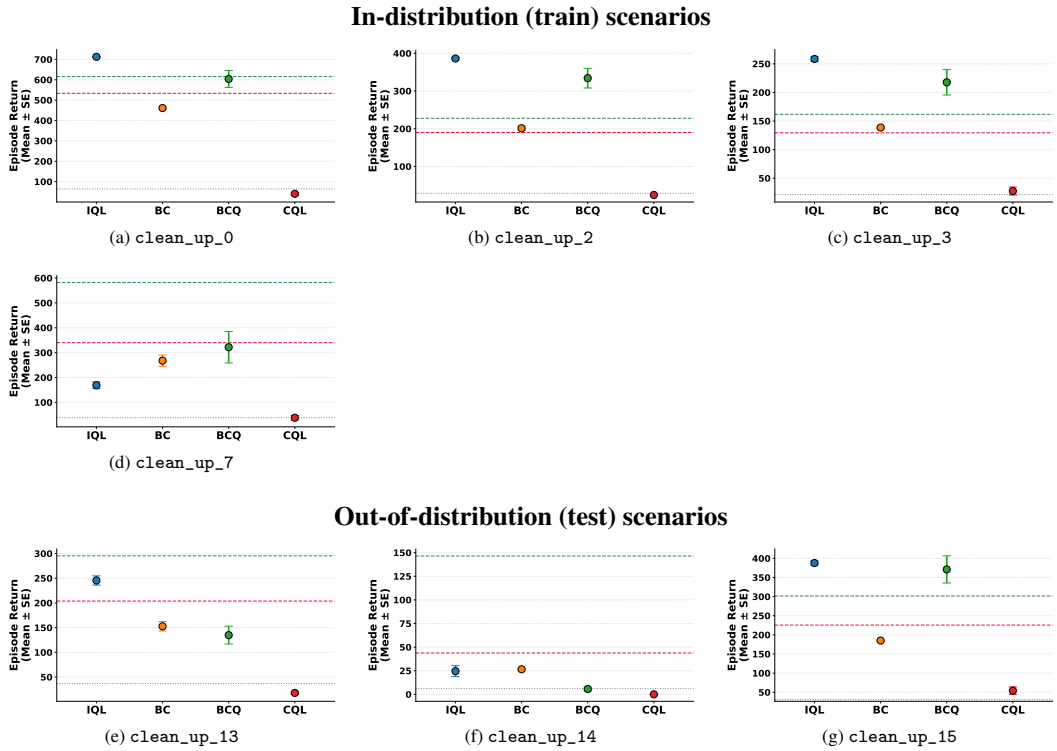


Figure 17: **Split U1 (Clean Up) — per-scenario Evaluation Setting 3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

E Per-scenario Evaluation Setting 3 eval returns by split

For each Evaluation Setting 3 train/test split we report the per-scenario final-eval returns in raw (un-normalised) return units, with one figure per split. Within each split's figure, in-distribution (train) scenarios are shown above the out-of-distribution (test) scenarios, so the in-/out-of-distribution generalisation gap is visible at a glance. Each panel shows the offline-RL algorithms as filled circles, where the marker is the mean across the seeds with data and the error bar is the standard error of that mean. Reference lines mark the dataset's mean episode return (red dashed), its p_{90} "best-in-dataset" ceiling (green dashed), and the random-policy baseline (grey dotted). Y-axes are in raw return units, scaled per-panel so within-scenario differences between algorithms remain legible across scenarios with very different return magnitudes.

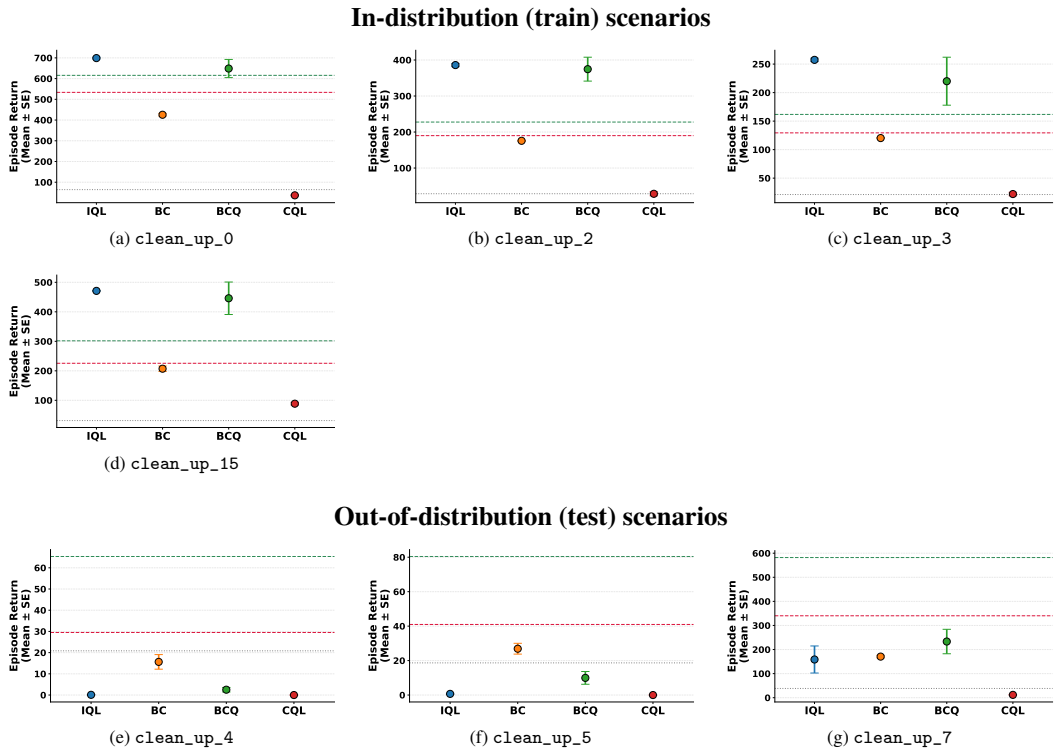


Figure 18: **Split U2 (Clean Up) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

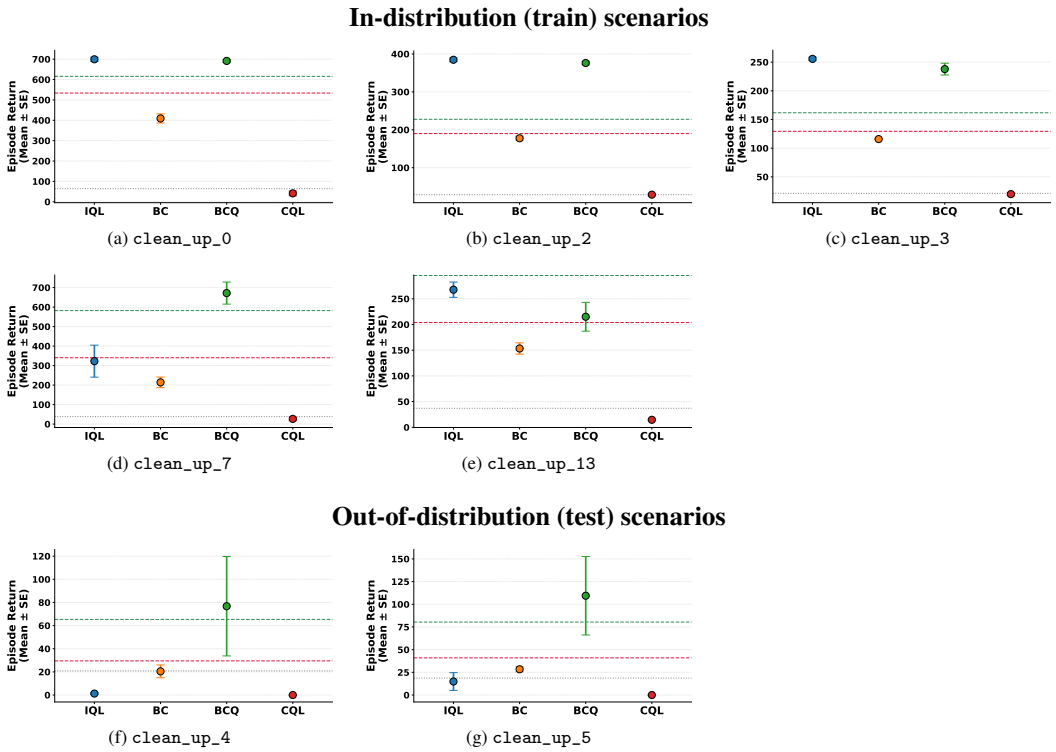
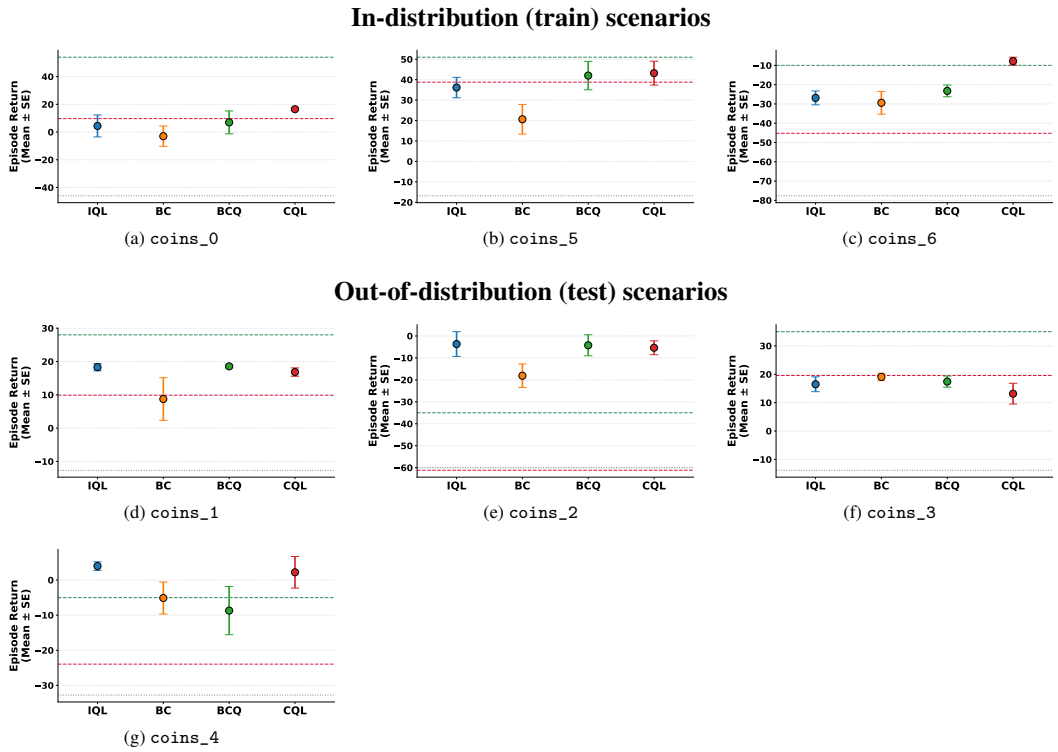


Figure 19: **Split U3 (Clean Up) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.



(a) coins_0 (b) coins_5 (c) coins_6

(d) coins_1 (e) coins_2 (f) coins_3

(g) coins_4

Figure 20: **Split C1 (Coins) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

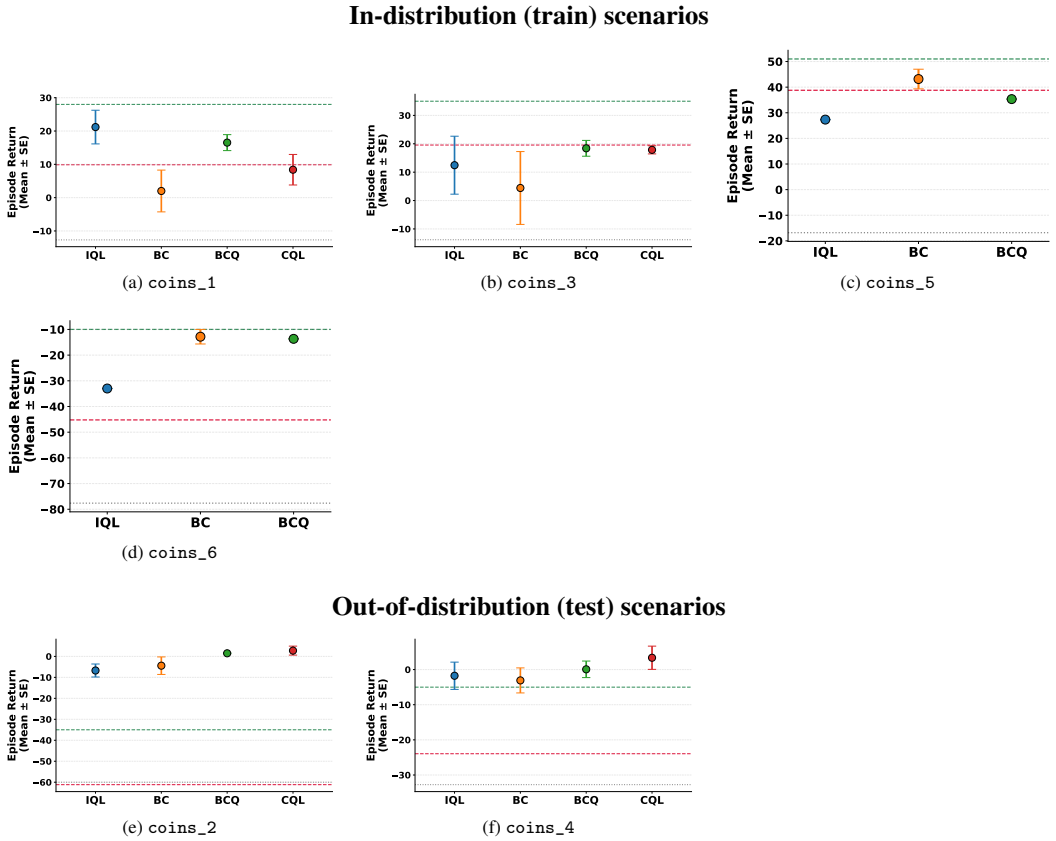


Figure 21: **Split C2 (Coins) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

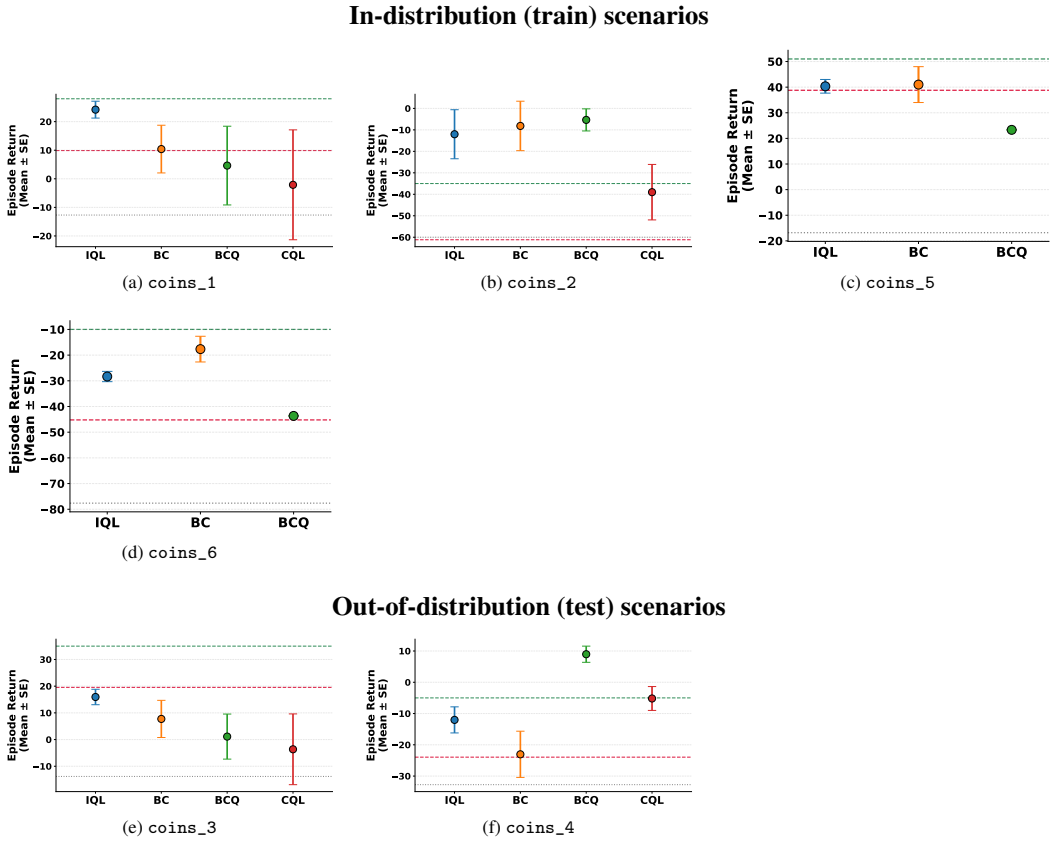
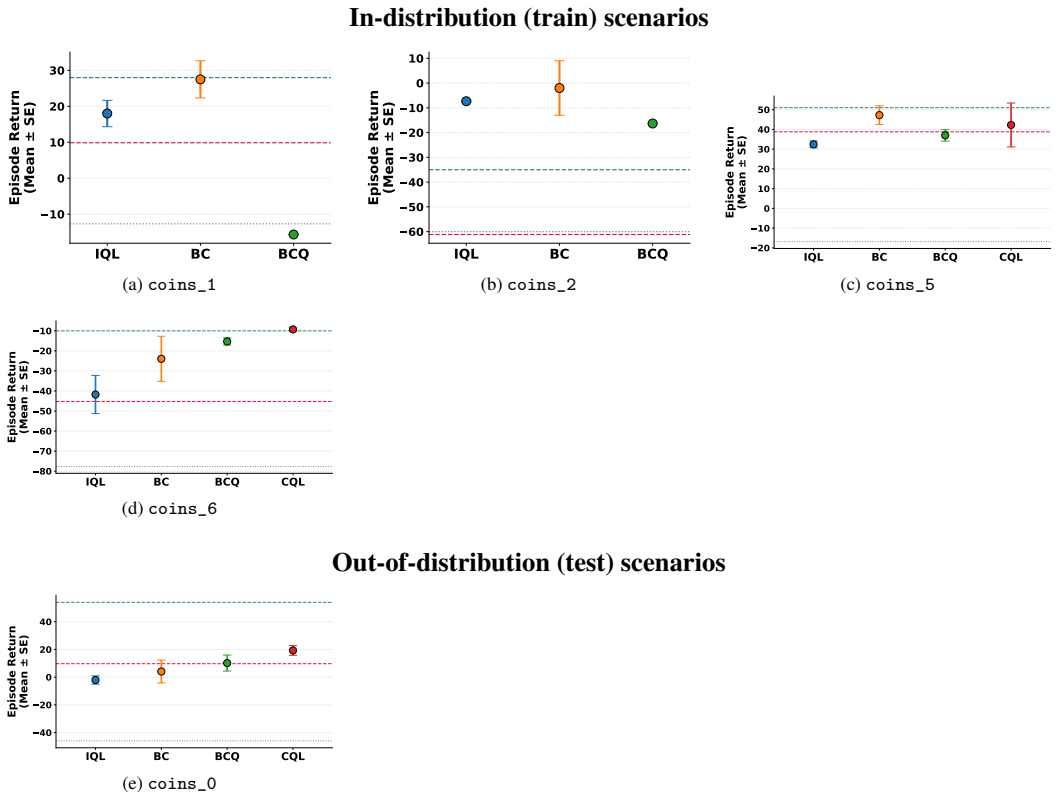


Figure 22: **Split C3 (Coins) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.



(a) coins_1 (b) coins_2 (c) coins_5 (d) coins_6 (e) coins_0

Figure 23: **Split C4 (Coins) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

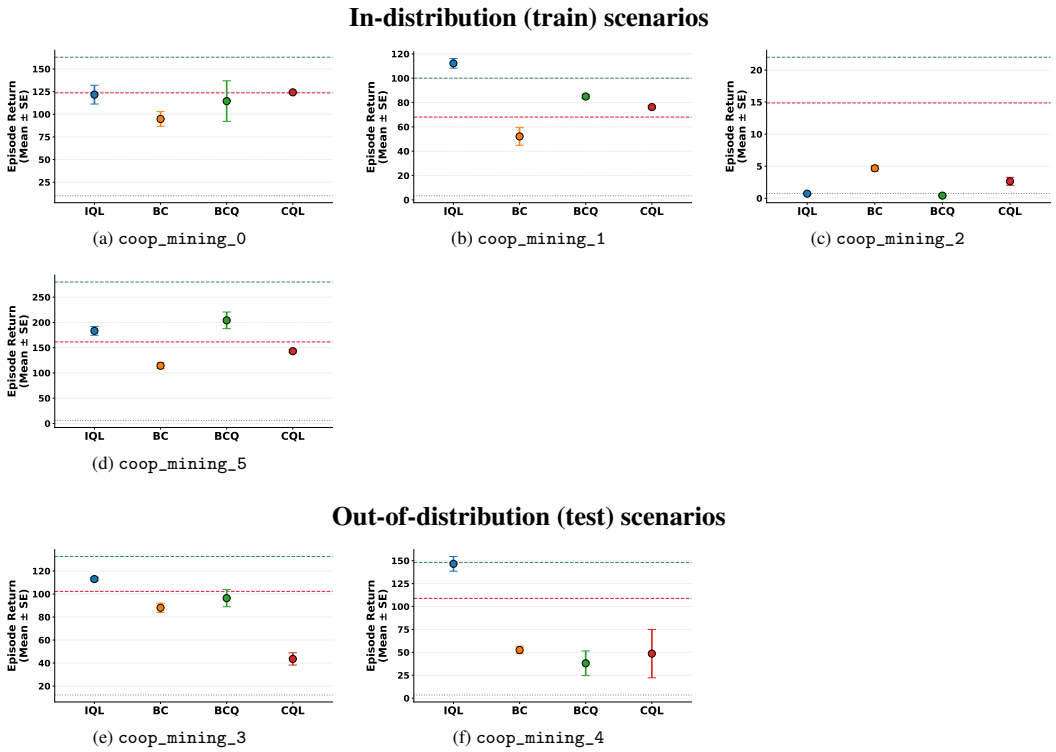


Figure 24: **Split M1 (Coop Mining) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

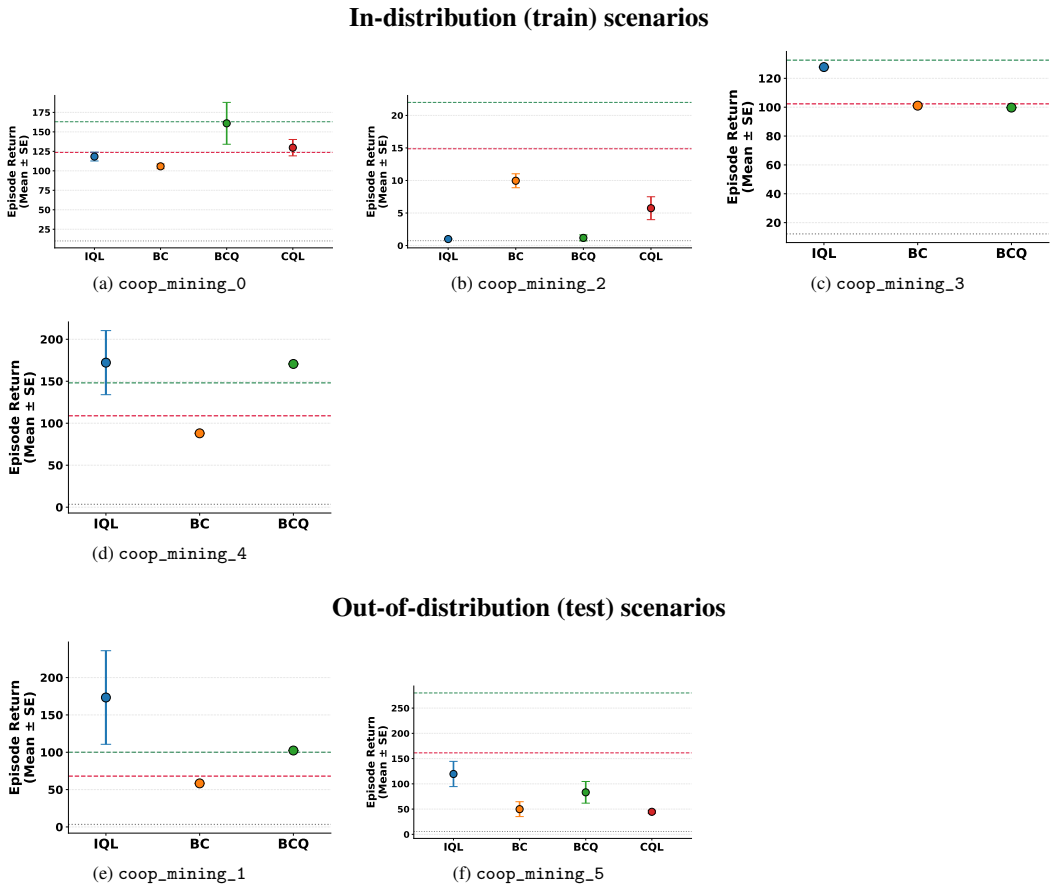


Figure 25: **Split M2 (Coop Mining) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

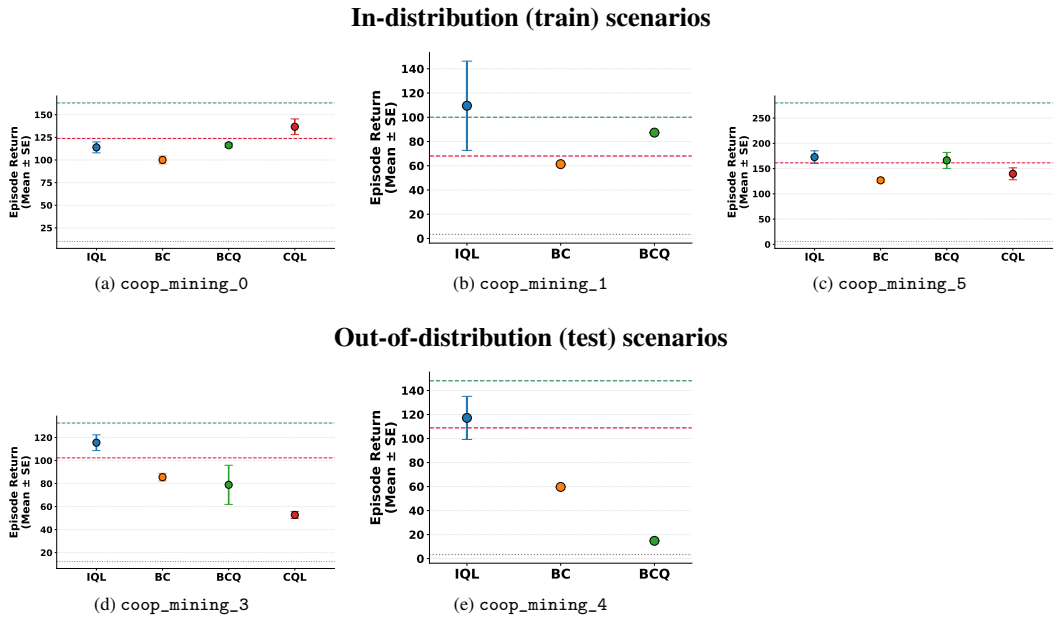


Figure 26: **Split M3 (Coop Mining) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

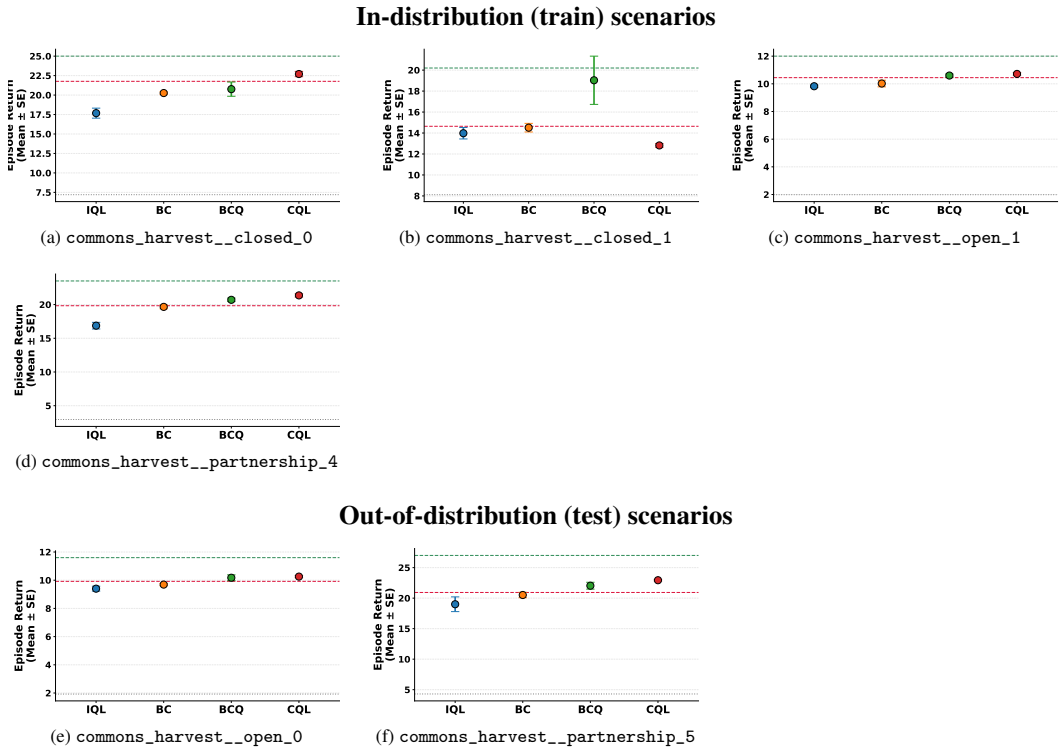


Figure 27: **Split H1 (Commons Harvest) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

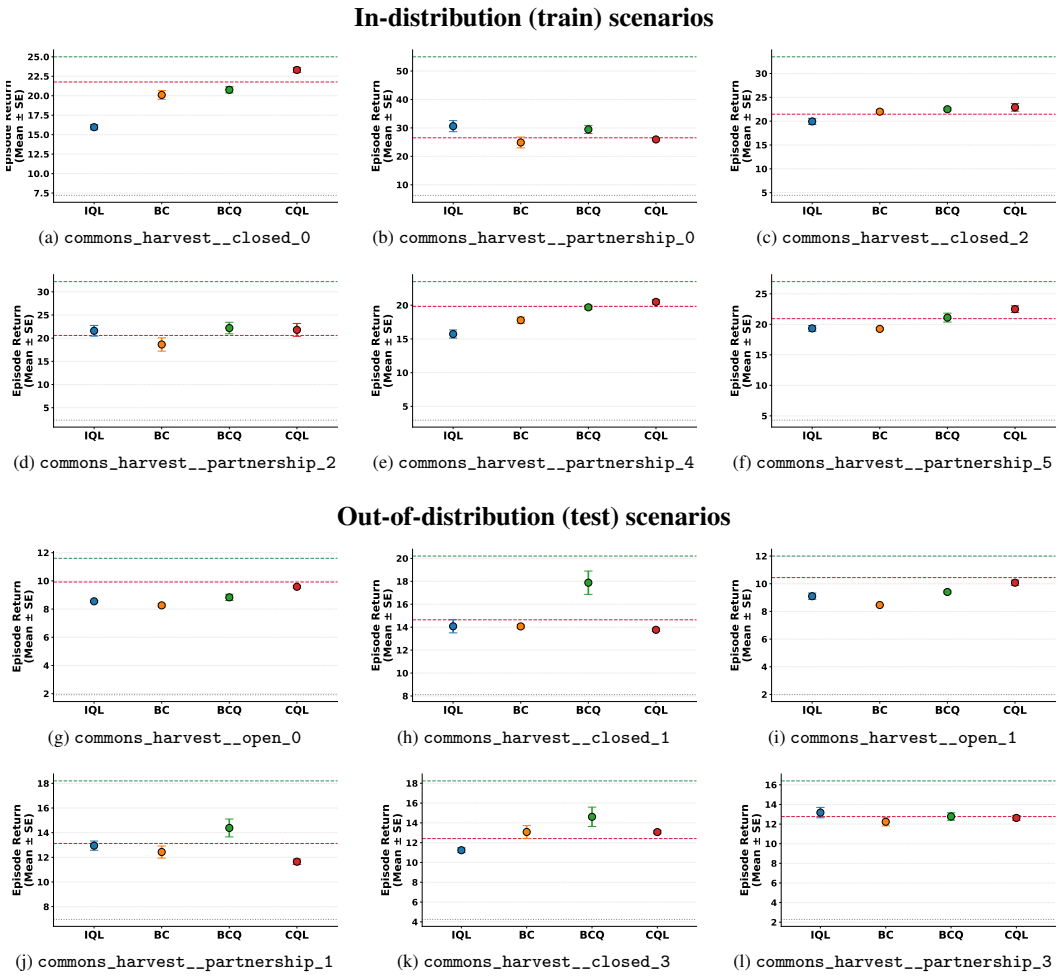


Figure 28: **Split H2 (Commons Harvest) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

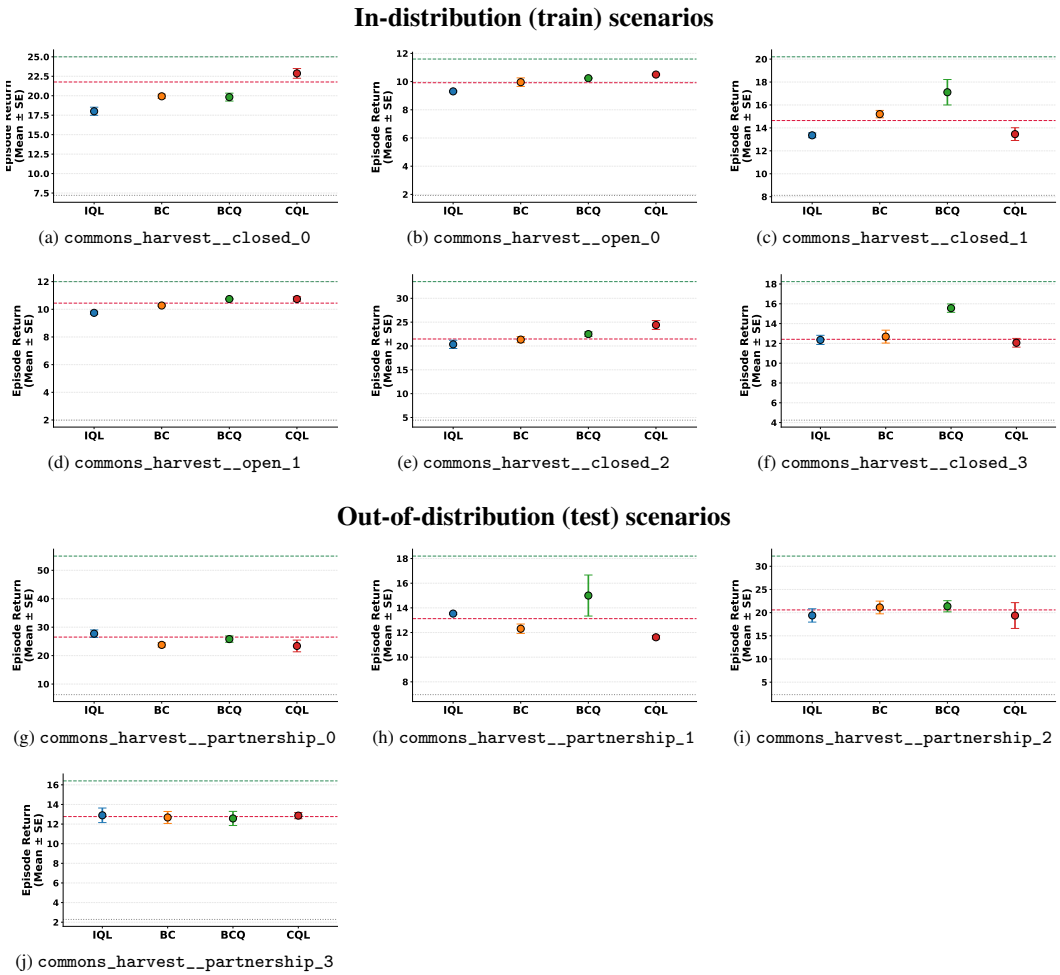


Figure 29: **Split H3 (Commons Harvest) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

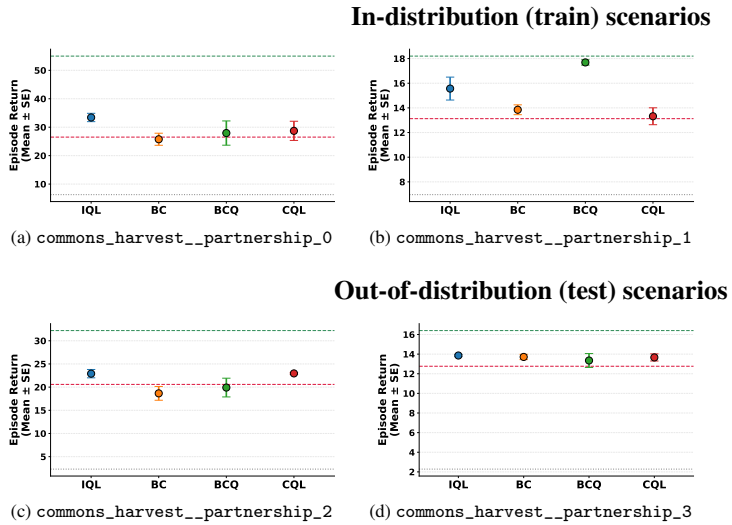


Figure 30: **Split H4 (Commons Harvest) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

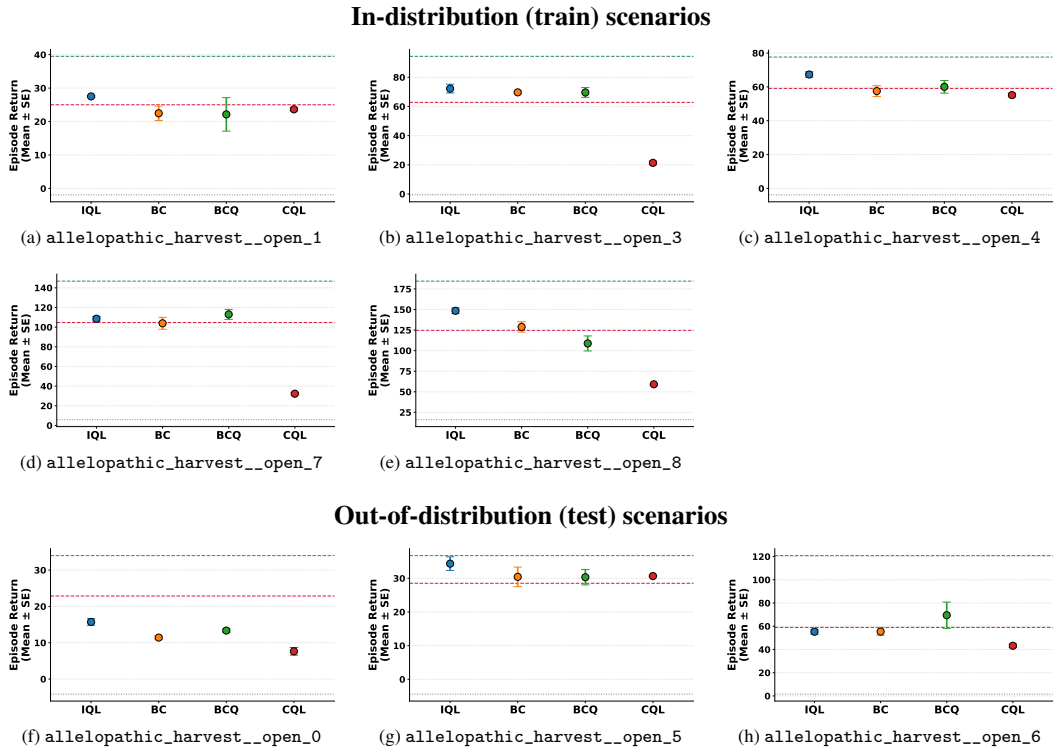


Figure 31: **Split A1 (Allelopathic Harvest) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

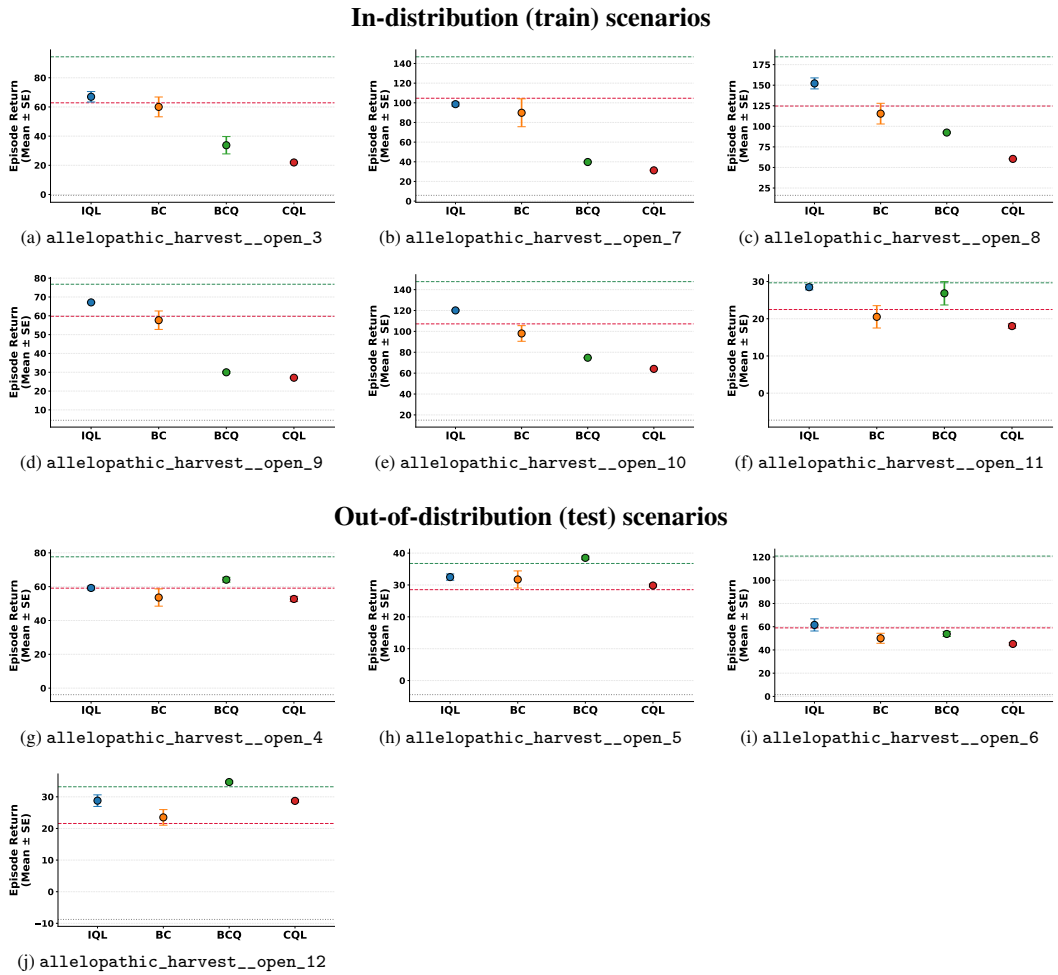


Figure 32: **Split A2 (Allelopathic Harvest) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

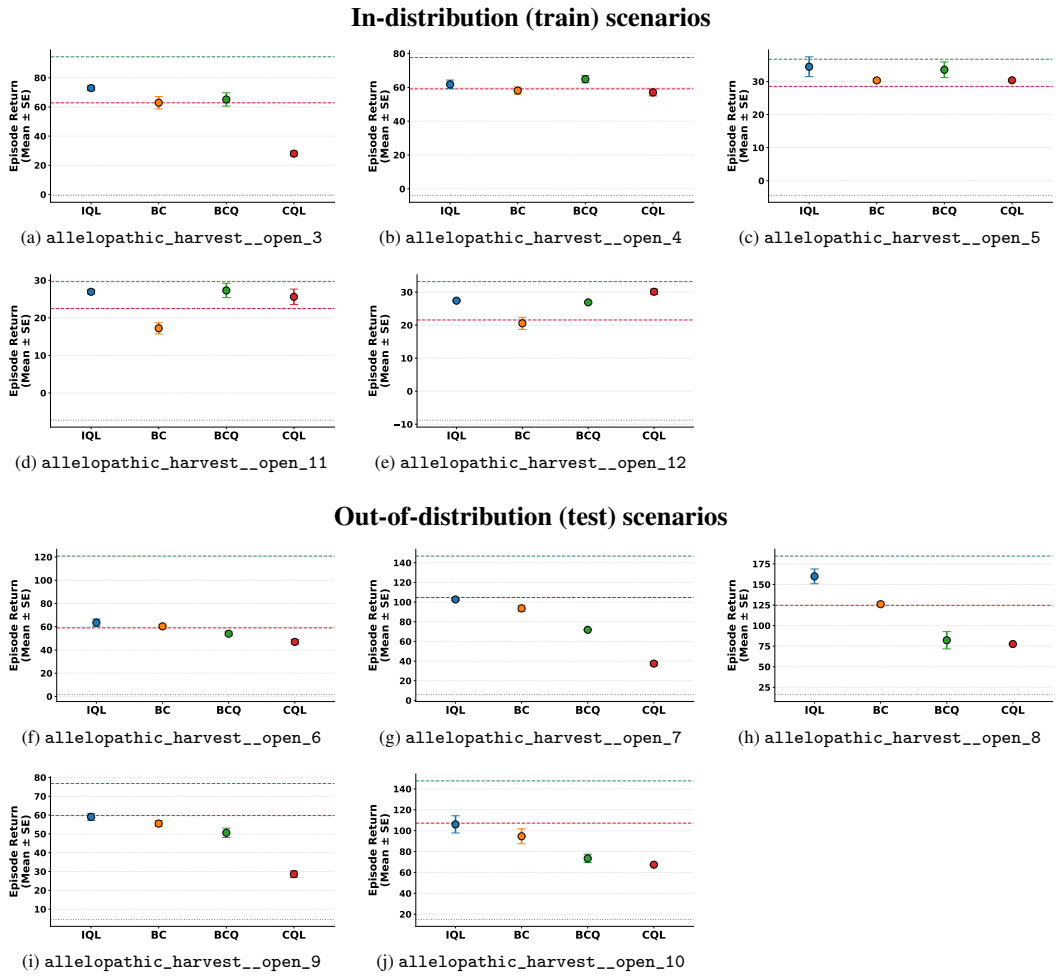
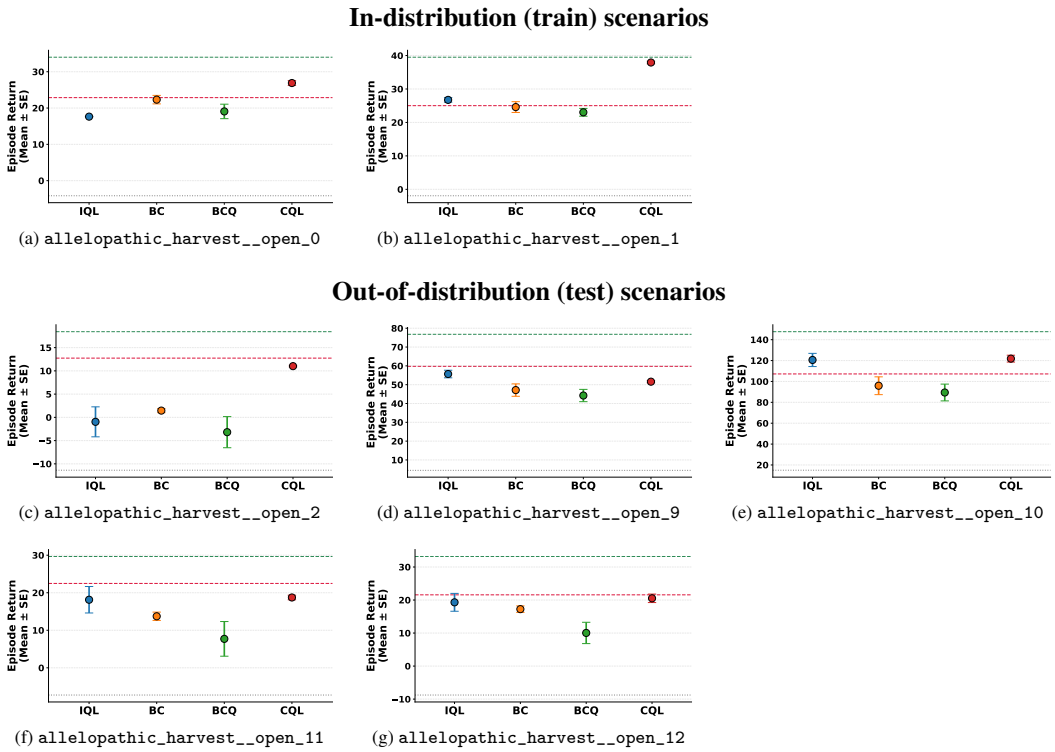


Figure 33: **Split A3 (Allelopathic Harvest) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.



(a) `allelopathic_harvest__open_0`
(b) `allelopathic_harvest__open_1`
(c) `allelopathic_harvest__open_2`
(d) `allelopathic_harvest__open_9`
(e) `allelopathic_harvest__open_10`
(f) `allelopathic_harvest__open_11`
(g) `allelopathic_harvest__open_12`

Figure 34: **Split A4 (Allelopathic Harvest) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

F Per-scenario background-agent return change

For each Evaluation 1 scenario we report the per-algorithm change in mean background-agent episode return induced by the trained focal policy, relative to a uniform-random focal policy ($\Delta = R_{\text{trained}} - R_{\text{random}}$). Background returns are read from the in-training eval summary (`eval/in_dist/<scen>/background_return_mean`) of the matching tier-1 source run. Negative values indicate the focal policy hurt the background population (exploitation); positive values indicate the focal policy helped them. Panels are grouped by substrate.

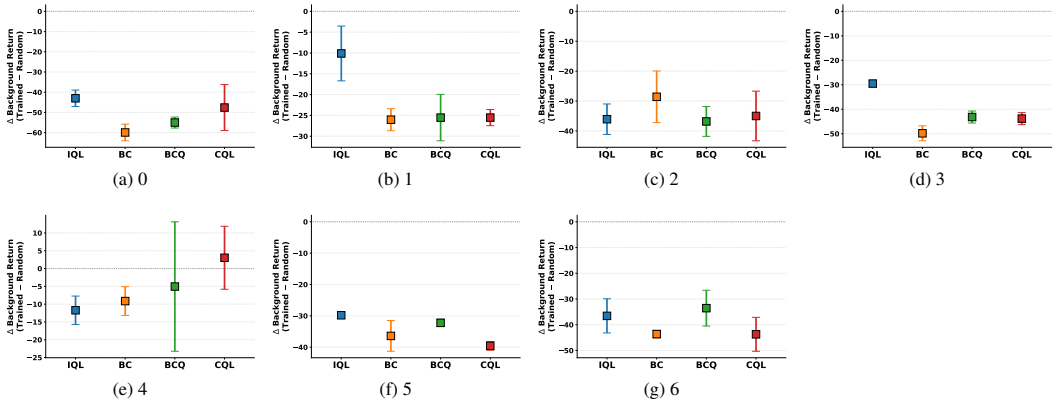


Figure 35: **Coins — background-agent return change.** For each scenario in Coins, one marker per algorithm showing the change in mean background-agent episode return induced by the trained focal policy, relative to a uniform-random focal policy ($\Delta = R_{\text{trained}} - R_{\text{random}}$). Background returns are read from the in-training eval logs. Negative values indicate the focal policy hurt the background population (exploitation); positive values indicate it helped them.

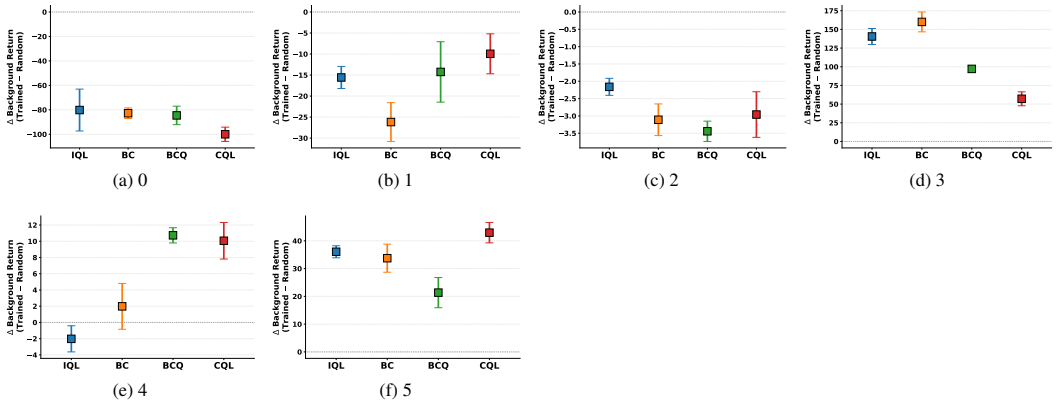
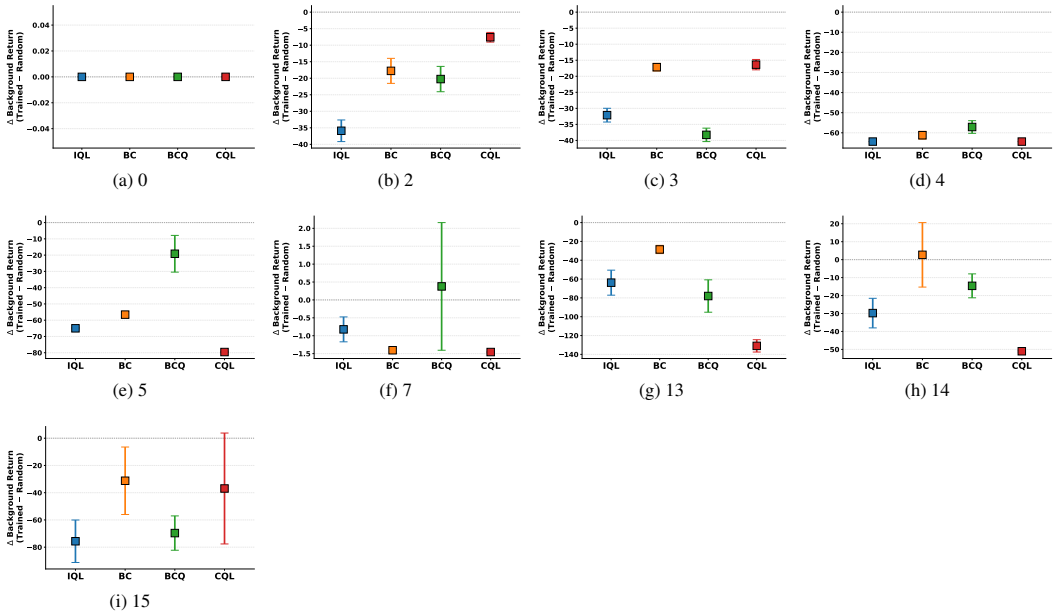


Figure 36: **Coop Mining — background-agent return change.** For each scenario in Coop Mining, one marker per algorithm showing the change in mean background-agent episode return induced by the trained focal policy, relative to a uniform-random focal policy ($\Delta = R_{\text{trained}} - R_{\text{random}}$). Background returns are read from the in-training eval logs. Negative values indicate the focal policy hurt the background population (exploitation); positive values indicate it helped them.

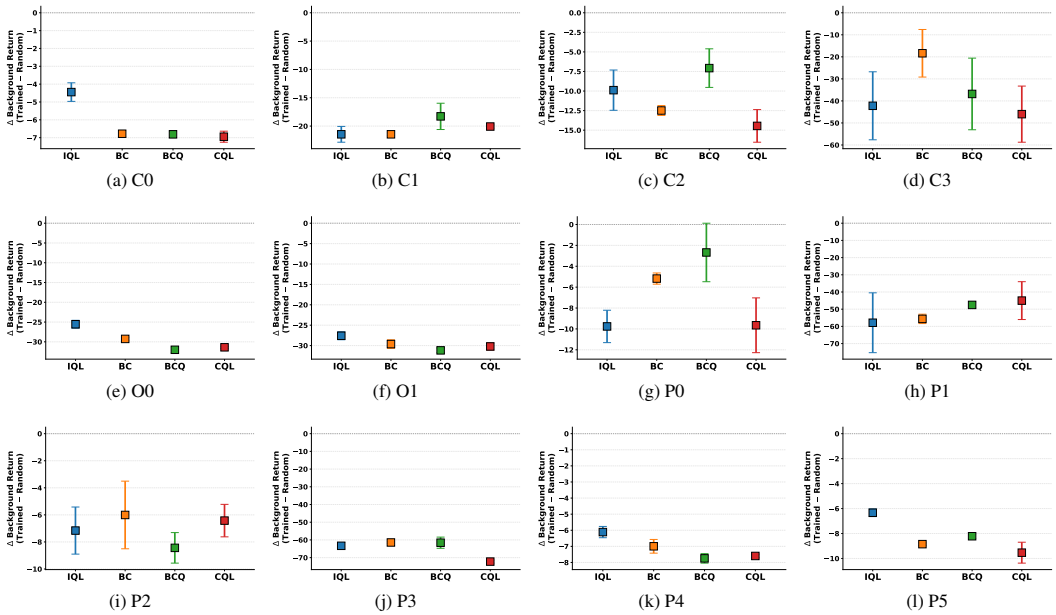


(a) 0 (b) 2 (c) 3 (d) 4

(e) 5 (f) 7 (g) 13 (h) 14

(i) 15

Figure 37: **Clean Up — background-agent return change.** For each scenario in Clean Up, one marker per algorithm showing the change in mean background-agent episode return induced by the trained focal policy, relative to a uniform-random focal policy ($\Delta = R_{\text{trained}} - R_{\text{random}}$). Background returns are read from the in-training eval logs. Negative values indicate the focal policy hurt the background population (exploitation); positive values indicate it helped them.



(a) C0 (b) C1 (c) C2 (d) C3

(e) O0 (f) O1 (g) P0 (h) P1

(i) P2 (j) P3 (k) P4 (l) P5

Figure 38: **Commons Harvest — background-agent return change.** For each scenario in Commons Harvest, one marker per algorithm showing the change in mean background-agent episode return induced by the trained focal policy, relative to a uniform-random focal policy ($\Delta = R_{\text{trained}} - R_{\text{random}}$). Background returns are read from the in-training eval logs. Negative values indicate the focal policy hurt the background population (exploitation); positive values indicate it helped them.

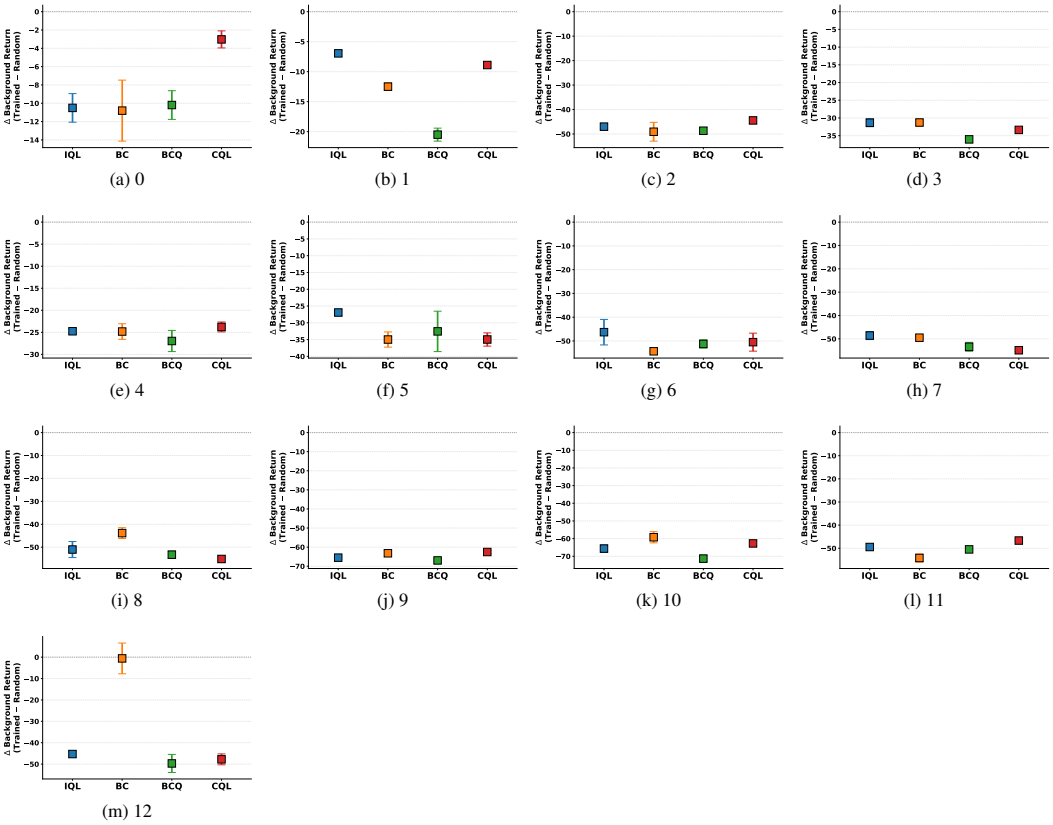


Figure 39: **Allelopathic Harvest — background-agent return change.** For each scenario in Allelopathic Harvest, one marker per algorithm showing the change in mean background-agent episode return induced by the trained focal policy, relative to a uniform-random focal policy ($\Delta = R_{\text{trained}} - R_{\text{random}}$). Background returns are read from the in-training eval logs. Negative values indicate the focal policy hurt the background population (exploitation); positive values indicate it helped them.

G Cross-substrate aggregate scores

Table 7 reports the full set of rliable headline estimators — Median, interquartile mean (IQM), mean, and optimality gap — pooled across every (substrate, scenario, seed) sample for Evaluation Settings 1 and 2. The compact optimality-gap view in the main text (Table 1) is the last row of each block here.

Table 7: **Cross-substrate aggregate scores — Evaluation 1 vs Evaluation 2.** Median, interquartile mean (IQM), mean, and optimality gap of the normalised final-eval return pooled across every (substrate, scenario, seed) sample, with stratified-bootstrap 95% CIs in brackets (rliable). Higher is better for the first three rows; optimality gap (↓). Best per row in bold.

Metric	IQL	BC	BCQ	CQL
Evaluation Setting 1 (single-scenario)				
Median	0.75 [0.73, 0.77]	0.72 [0.68, 0.73]	**0.80 [0.77, 0.83]**	0.63 [0.60, 0.66]
IQM	0.73 [0.71, 0.74]	0.69 [0.68, 0.71]	**0.80 [0.78, 0.82]**	0.63 [0.62, 0.65]
Mean	0.78 [0.75, 0.81]	0.66 [0.65, 0.67]	**0.89 [0.86, 0.92]**	0.58 [0.56, 0.59]
Optimality Gap	0.30 [0.28, 0.33]	0.35 [0.33, 0.36]	**0.26 [0.24, 0.27]**	0.43 [0.42, 0.44]
Evaluation Setting 2 (multi-scenario, scenario label withheld)				
Median	0.72 [0.71, 0.74]	0.68 [0.65, 0.69]	**0.73 [0.70, 0.78]**	0.58 [0.52, 0.63]
IQM	**0.73 [0.72, 0.75]**	0.66 [0.65, 0.67]	**0.73 [0.71, 0.77]**	0.57 [0.54, 0.58]
Mean	0.78 [0.77, 0.80]	0.66 [0.65, 0.68]	**0.79 [0.75, 0.83]**	0.55 [0.53, 0.56]
Optimality Gap	0.31 [0.29, 0.32]	0.36 [0.35, 0.37]	**0.29 [0.27, 0.31]**	0.49 [0.48, 0.51]