MOLTEN POT: EVALUATIONS & DATASETS FOR SOCIAL OFFLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

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 and datasets 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 with the scenario label withheld, so a single policy must learn from heterogeneous, unlabelled partner behaviour. Setting 3 evaluates zero-shot social generalisation through disjoint train/test splits that target specific social shifts. Finally, we provide four offline RL baselines (IQL, BC, BCQ, CQL) as a reliable starting point for future methods. They leave substantial headroom on our protocol, and we invite the community to adopt the benchmark and close the gap. The goal of Molten Pot is to establish 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 (Tomasello et al., 2005; Baker et al., 2017). 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 (Baker et al., 2009). This capacity for social reasoning in novel settings is crucial for trusting agents to act in the real world (Dafoe et al., 2020). 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 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 (Gottesman et al., 2018). In each case the data is plentiful, the environment is social, and the agent has no opportunity to explore online (Levine et al., 2020; Gottesman et al., 2018). 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 (Fu et al., 2020), RL Unplugged (Gulcehre et al., 2020), 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. These benchmarks stress distributional shift between the behaviour policy π^β and the learned policy π^θ, and the conservatism required to avoid over-

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

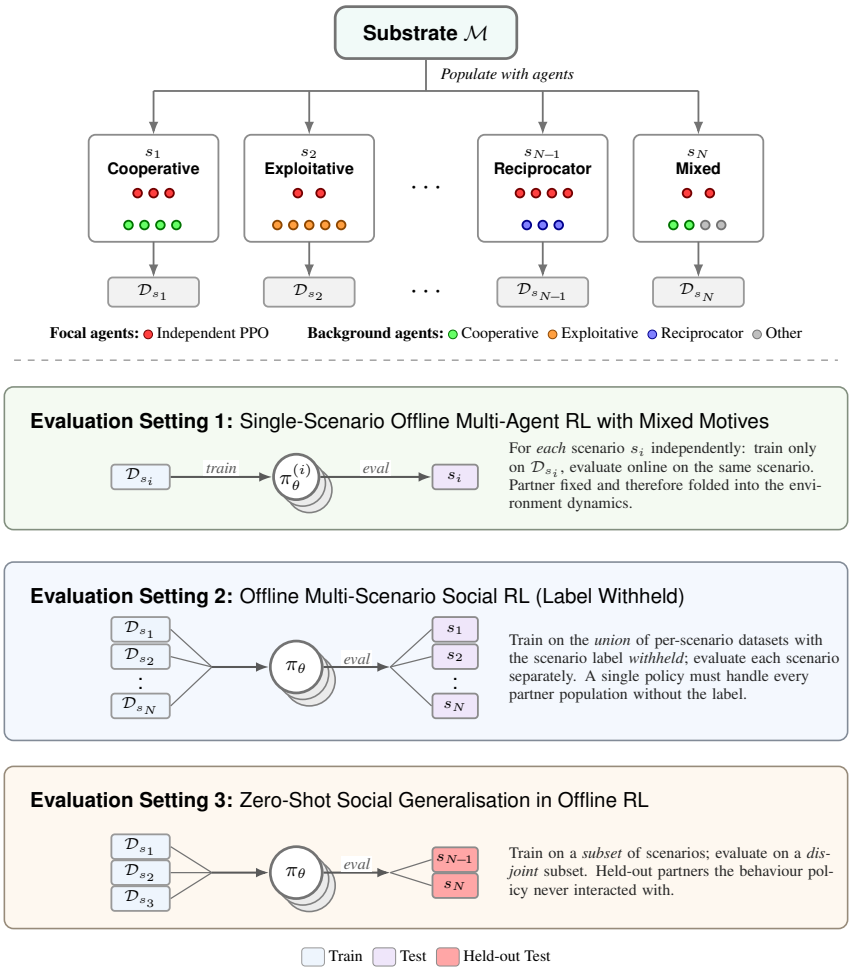


Figure 1: **Molten Pot at a glance.** *Top:* a substrate $\mathcal{M}$ paired with different background populations $\pi_s^{\mathcal{B}}$ defines a family of scenarios $s_i = \langle \mathcal{M}, \pi_{s_i}^{\mathcal{B}} \rangle$. Every scenario hosts multiple *focal* agents, each controlled by its own 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; Evaluation Setting 3 trains on a subset of scenarios and evaluates on a disjoint held-out set of scenarios.

estimating out-of-distribution actions, but *not* the social reasoning challenges that arise when the environment contains other adaptive agents. The standard offline RL question of whether we can learn a robust policy from historical logs then becomes whether we can learn one from logs of interactions with a *population of co-players with mixed motives*, and whether that *policy generalises to unseen players*.

Offline multi-agent RL benchmarks move closer but do not reach this goal. OG-MARL and its extensions (Formanek et al., 2023; 2026) target offline Multi-Agent RL (MARL) in fully cooperative environments such as SMAC (Samvelyan et al., 2019) and MAMuJoCo (Peng et al., 2021), where the central challenge is coordination (Tilbury et al., 2024) 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 (Leibo et al., 2021) 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 remains unexplored, namely: learning mixed-motive social policies from fixed datasets and evaluating on these substrates.

We introduce *Molten Pot*, an evaluation protocol with datasets for offline mixed-motive, social reinforcement learning. Throughout, we use *social robustness* to mean the ability of an individually-rational focal policy to adjust appropriately to variation in co-player behaviour, in the mixed-motive sense of Melting Pot (Leibo et al., 2021), where *social* refers to interaction among agents with non-identical incentives. It makes no claim about group welfare, fairness, or prosociality, all of which lie beyond the individual-return metric and which we discuss in section 6. 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 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 (Formanek et al., 2023) 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, so policies must handle heterogeneous partner behaviour without explicit context, whether by partner inference, by learning a robust generalist strategy, or otherwise. *Evaluation Setting 3* holds out scenarios exhibiting specific classes of social shift, including convention mismatch, reciprocity threshold, punishment capability, deployment role, and latent within-episode partner type, and tests generalisation to partner behaviours the training data never contained. The settings are not a difficulty ladder: 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 (Hughes et al., 2024) 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.

Contributions. This work makes three contributions.

- To our knowledge, Molten Pot is the first offline MARL work to propose empirical evaluations for mixed-motive social dilemmas, a setting existing offline benchmarks leave out.[1]
- We release over one terabyte of high-quality trajectory data, logged from Melting Pot training runs, on Hugging Face.[2] Offline MARL is held back by how little diverse data it has: what exists is almost exclusively fully cooperative SMAC and MAMuJoCo (Formanek et al., 2026).
- We provide a structured evaluation protocol whose three settings each probe a different element of social robustness, together with guidelines for statistically sound algorithmic comparison (Agarwal et al., 2021; Gorsane et al., 2022) and four stable baselines released on GitHub.[3]

2 Preliminaries

2.1 Partially Observable Markov Games

Following Leibo et al. (2021), 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 (Gronauer & Diepold, 2022; Hansen et al., 2004)

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

[1] Interactive visualization: `https://frmjua.github.io/moltenpot`

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

[3] Code: `https://github.com/frmjua/moltenpot/`

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. (2021), 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.

2.3 Scenarios

A *scenario* is a pair $s = \langle \mathcal{M}, \pi_s^{\mathcal{B}} \rangle$ comprising a substrate $\mathcal{M}$ 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 $\mathcal{M}$ 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 (Schulman et al., 2017): 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 (Gronauer & Diepold, 2022) 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. (2021) on the same scenarios, confirming that the behaviour policy is a reasonable reference against which to normalise offline methods. Following the dataset-transparency recommendation of Formanek et al. (2026), the aggregated return distribution per substrate is shown in Figure 3 and the per-scenario breakdown in Appendix E.

3 Related Work

Standard single-agent offline RL benchmarks—D4RL (Fu et al., 2020) and RL Unplugged (Gulcehre et al., 2020)—fix per-task behaviour distributions so that conservative algorithms such as BCQ (Fujimoto et al., 2019), CQL (Kumar et al., 2020), and IQL (Kostrikov et al., 2021) can



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

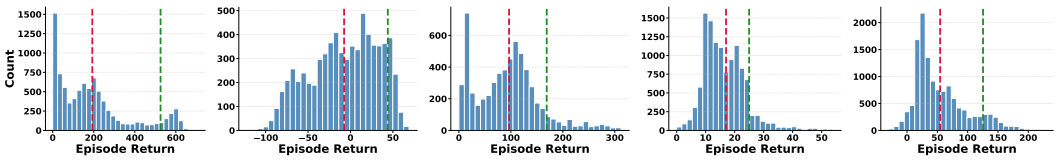
Figure 2: **The five Molten Pot substrates.** Representative rendered frames from DeepMind's Melting Pot (Leibo et al., 2021) covering a range of mixed-motive game structures: common-pool resource dilemmas, public-goods games, dyadic social dilemmas, partner-choice coordination, and convention adoption. 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.

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 (Formanek et al., 2023) and its revision (Formanek et al., 2026), which inherit the cooperative assumption from SMAC (Samvelyan et al., 2019) and MAMuJoCo (Peng et al., 2021) and therefore do not evaluate partner reasoning; offline-MARL algorithmic work has likewise concentrated on cooperative coordination (Tilbury et al., 2024; Hedman et al., 2026; Barde et al., 2023), multi-agent conservatism such as OMAR (Pan et al., 2021) and counterfactual conservative Q-learning (Shao et al., 2023), model-based approaches (Barde et al., 2023; Willemsen et al., 2021), and offline-to-online transfer (McInroe et al., 2023). These methods are largely built on centralised training with decentralised execution (CTDE), which presumes a shared team reward and is therefore incompatible with the individual incentives of the mixed-motive setting, so they do not apply to Molten Pot. The mixed-motive evaluation protocol Molten Pot adopts originates with sequential social dilemmas (Lerer & Peysakhovich, 2018) and Melting Pot (Leibo et al., 2021; Agapiou et al., 2022), whose focal/background split scores generalisation to unseen co-players, and which has since been reimplemented for fast online training in SocialJax (Mizuta et al., 2025); in the mixed-motive online setting, Ndousse et al. (2020) further show that social observation can drive agents to learn from one another. Partner handling itself is evaluated in cooperative form by ZSC-Eval (Wang et al., 2024) and the ad-hoc-teamwork literature, with Mon-Williams et al. (2025) studying when recurrent agents form partner models. Applied mixed-motive MARL has also begun to make its way into human-compatible self-driving (Cornelisse & Vinitsky, 2024; Cusumano-Towner et al., 2025).

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.



(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 E.

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:

$$J_{\text{I}}(\pi_\theta;\, s) \;=\; J_s(\pi_\theta), \qquad s \in \mathcal{S}_{\text{all}}. \tag{2}$$

What this setting evaluates. Because s is fixed, $\pi_s^{\mathcal{B}}$ is a fixed feature of the environment dynamics and the partner population does not vary across episodes. Setting 1 therefore mirrors the standard single-task offline POSG from Formanek et al. (2023), 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 (Formanek et al., 2023; 2024). 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: Offline Multi-Scenario Social RL with Withheld Scenario Labels

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

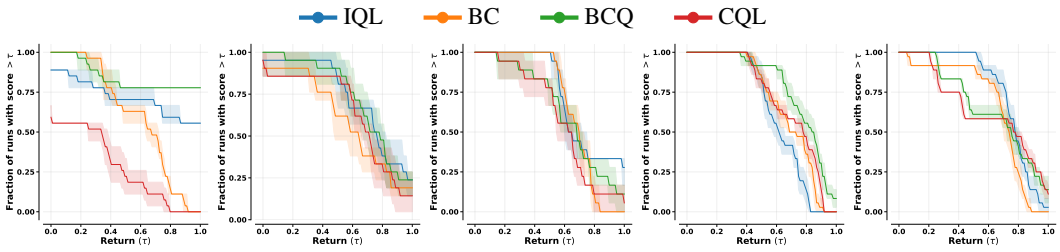
$$J_{\text{II}}(\pi_\theta) \;=\; \mathbb{E}_{s\sim P^{\text{te}}(\mathcal{S}_{\text{sub}})}\big[\, J_s(\pi_\theta)\,\big], \tag{3}$$

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 pools heterogeneous scenarios with the scenario label withheld, so a single policy must be learned from mixed, unlabelled partner behaviour. It has several additional challenges which are not present in Setting 1: multi-scenario (multi-task) learning, more diverse data, conflicting best responses across scenarios, and plausibly the need to infer partner behaviour from within-episode history (an idealised Bayesian view of this last point is given in Appendix B). Poor performance can equally reflect standard offline-RL errors such as extrapolation error, insufficient support, and conservative bias. To our knowledge no existing benchmark offers this kind of offline multi-scenario social setting.

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 (Agarwal et al., 2021; Gorsane et al., 2022): 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 $P^{\text{tr}}(s) \neq P^{\text{te}}(s)$ with disjoint scenario support, so test partners lie outside the training scenario set (in the idealised Bayesian view of Appendix B, the partner posterior is miscalibrated by construction). 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. Setting 3 asks whether policies learned from a subset of scenarios transfer to partner populations entirely absent from the training distribution. Each train/test split targets 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. As in Setting 2, poor held-out performance may also reflect multi-task learning, limited data coverage, or model capacity rather than partner handling alone. Table 4 in Appendix A lists the eighteen splits across the five substrates and characterises each shift in detail.

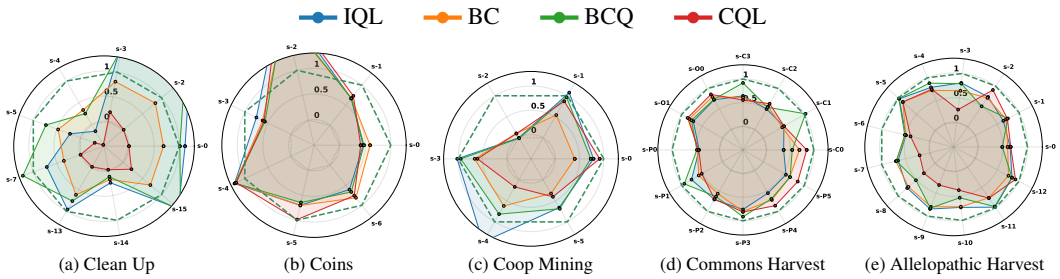
5 Baseline Results

We evaluate four baseline implementations on Evaluation Settings 1–3: behavioural cloning (BC, Pomerleau, 1991), Batch-Constrained Q-Learning (BCQ, Fujimoto et al., 2019), Implicit Q-Learning (IQL, Kostrikov et al., 2021), and Conservative Q-Learning (CQL, Kumar et al., 2020). Every algorithm's policy and Q-function carries a recurrent GRU (Cho et al., 2014) layer over the focal agent's interaction history, which handles partial observability and can in principle also support partner inference (Mon-Williams et al., 2025). We use single-agent offline algorithms because existing offline MARL methods rely on centralised training with a shared team reward, which is ill-suited to mixed-motive substrates with distinct per-agent rewards (Section 3); independent-agent baselines are moreover competitive with, and often outperform, CTDE methods on standardised offline MARL benchmarks (Formanek et al., 2024). Each baseline is a single implementation trained for 20,000 gradient updates with three seeds and a deliberately modest hyperparameter budget tuned on a single scenario (Appendix C).

Normalisation. For each scenario we report the normalised final eval return $\tilde{J}_s = (R_s - R_{\text{random},s})/(R_{p90,s} - R_{\text{random},s})$ (Fu et al., 2020), where $R_{\text{random},s}$ is the mean return of a uniform-random focal policy on scenario s and $R_{p90,s}$ the 90th-percentile episode return in the offline dataset for s, our proxy for expert play. So 0 corresponds to a random policy and 1 to the "best-in-dataset" (top-decile) episode; every result figure marks $\tilde{J} = 0$ (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



(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} reference and the dotted grey circle marks the random-policy baseline. Dents toward the centre mark scenarios with low normalised return.

computed by MARL-evals utilities (Gorsane et al., 2022), with shaded stratified-bootstrap confidence intervals (Agarwal et al., 2021). Profile shapes differ across the baselines. On most substrates BC's is a near-flat plateau followed by a sharp drop at a substrate-specific threshold, i.e. BC scores in a narrow band on most scenarios, whereas the offline RL methods (IQL, BCQ, CQL) trace more gradually decreasing curves: they fall below BC's on scenarios inside its plateau but retain non-zero mass beyond BC's drop-off, including scenarios well above the dataset p_{90} as well as scenarios near random. We do not attribute this spread to the algorithms themselves, since hyperparameters were tuned on a single scenario per substrate (Appendix C). In Appendix G 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 how each baseline's return varies across scenarios. 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, marking scenarios that remain hard for all four baselines.

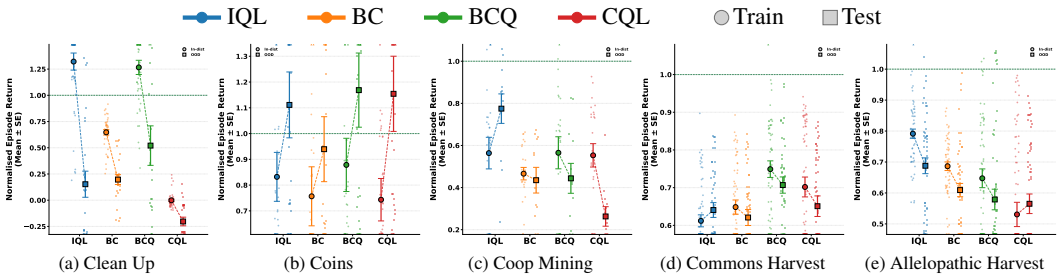
Setting 1 vs Setting 2. Table 1 compares the two settings by their optimality gap, the mean shortfall of the normalised return below the p_{90} reference, so lower is better. Values are pooled across every (substrate, scenario, seed) sample, with stratified-bootstrap 95% confidence intervals computed with RLiable (Agarwal et al., 2021; Gorsane et al., 2022). The gap is equal or slightly larger in Setting 2 for all four baselines, most clearly for CQL, while the intervals overlap or touch for IQL, BC and BCQ. This is the direction we would expect: Setting 2 adds the challenges described in Section 4.2, namely learning from several heterogeneous scenarios at once, more diverse data, conflicting best responses across scenarios, and unlabelled partner populations.

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

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 F.

filled square) evaluations shown side by side. Within every substrate the in-distribution markers sit above their out-of-distribution counterparts: out-of-distribution scores are lower for every baseline and on several substrates approach the random-policy reference. The gap size varies by algorithm and substrate. Punishment- and role-shifts on Commons Harvest produce the largest drops, and Coins the smallest. In- and out-of-distribution scores are computed on different scenarios, so the gap also reflects differences in scenario difficulty rather than shift alone. None of the four baselines closes the in/out gap, leaving headroom for future methods. Per-split, per-scenario raw-return breakdowns appear in Appendix F.

6 Conclusion

We introduced Molten Pot, 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), multi-scenario learning under withheld scenario labels (Setting 2), and zero-shot generalisation across disjoint train/test partner populations (Setting 3). The four baselines (IQL, BC, BCQ, CQL) are a reliable starting point for future work: they leave headroom to the p_{90} reference on Settings 1 and 2, and their performance drops out-of-distribution on Setting 3. Progress on these settings may need methods beyond scenario-agnostic conservatism, such as multi-task offline learning, partner representations that extrapolate, or memory architectures that track partner type. We see offline MARL as held back less by a shortage of algorithmic ideas than by the narrow, almost exclusively cooperative evaluations available to test them. Molten Pot is intended as a catalyst for moving past that default, away from the fully cooperative, CTDE setting that has shaped the field (Formanek et al., 2024) and towards decentralised methods that cope with misaligned incentives, with each setting a step in that direction. We provide the datasets, splits and harness to measure that progress, and hope the community will adopt them as a foundation for the next generation of multi-agent offline RL.

Limitations Molten Pot scores a focal policy by its own expected return, following the evaluation protocol of Melting Pot (Leibo et al., 2021). Because the background populations are fixed, a focal policy can raise its individual return by adapting to the specific behaviour of the shipped co-players, and in some scenarios the highest-return adaptation is to exploit them, for example by starving them of resources or by provoking their scripted punishment mechanics for personal gain. This follows from measuring individual best-response in mixed-motive games and is a property of the evaluation target rather than an artefact of the datasets: a high score certifies individual-return competence only, and is not evidence of prosocial, welfare-improving, or fair behaviour. We report the change in background-agent return induced by the trained focal policy (Appendix G), so a policy that achieves high return at the expense of its co-players is visible in the logged metrics rather than hidden. These diagnostics surface exploitation but do not remove it from the objective; a welfare-aware or fairness-aware evaluation target remains an important extension that we leave to future work.

REFERENCES

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.

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.

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.

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.

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.

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.

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

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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

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.

Vladislav Kurenkov and Sergey Kolesnikov. Showing your offline reinforcement learning work: Online evaluation budget matters. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

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.

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

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

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.

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

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.

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

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.

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.

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.

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

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.

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

Jianzhun Shao, Yun Qu, Chen Chen, Hongchang Zhang, and Xiangyang Ji. Counterfactual conservative q learning for offline multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

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.

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.

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.

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: Partner Inference and Bayes Optimality: An Idealised View
- Appendix C: Hyperparameters
- Appendix D: Compute
- Appendix E: Per-scenario return histograms
- Appendix F: Per-scenario Evaluation Setting 3 eval returns by split
- Appendix G: Per-scenario background-agent return change
- Appendix H: Cross-substrate aggregate scores
- Appendix I: LLM Usage

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 (Lerer & Peysakhovich, 2018)	2	Reciprocity and adaptation to partner policy in a mixed-motive dyadic dilemma.
Coop Mining	6	Trust formation, partner choice.
Commons Harvest (Perolat et al., 2017)	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.

Table 3 (continued)

Substrate	Sc.	Scenario Type	Description
Commons Harvest: Open	0	R CG ANTI EN	Focals are resident and visited by two bots who zap and harvest unsustainably.
	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	Sparsely populated 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 PC → RC TH	1, 3, 5, 6	2, 4	Train on generous reciprocators and unconditional partners; test on harsh (low-threshold) reciprocators.
	C3	RC PC → RC EN	1, 2, 5, 6	3, 4	Train on normal-punishment reciprocators and unconditional partners; test on strong-punishment reciprocators.
	C4	RC PC → PC LT	1, 2, 5, 6	0	Train on fixed-type partners; test on a partner whose type (pure cooperator or pure defector) is latent.
Coop Mining	M1	V → R	0, 1, 2, 5	3, 4	Transfer from visitor-mode response to resident-mode competition.
	M2	TR PC → PC LT	0, 2, 3, 4	1, 5	Train on homogeneous partner populations; test on mixed/generalist scenarios, one with a latent partner type.
	M3	V PC → R PC	0, 1, 5	3, 4	Train on abundant/flexible visitor scenarios; test on competitive resident scenarios where access itself is the bottleneck.
Commons Harvest	H1	ANTI → ANTI EN	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 → TR PRO EN	Partnership: 0, 1	Partnership: 2, 3	Trust calibration when good partners give way to partners who still harvest sustainably but withhold trust and punish.
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 Partner Inference and Bayes Optimality: An Idealised View

This appendix gives an optional formal characterisation of one idealised way to view the problem in Evaluation Settings 2 and 3, where the scenario label is withheld and partner populations are never identified by label. It is not needed to follow the protocol or results in the main text, and nothing in our benchmark or baselines assumes that a learner represents any quantity below. We do not claim that partner inference is necessary for good performance on these settings, or that the benchmark isolates it.

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 (Baker et al., 2009). We can formalise this structure. Writing $h_t = (o_0, a_0, r_0, \ldots, o_t)$ for the focal history (and $h_t^{\mathcal{B}}$ for the background agents' history, which the focal agent does not observe), the focal-perspective transition under scenario s is obtained by marginalising over background actions:

$$p\big(o_{t+1}, r_t \,\big|\, h_t, a_t;\, s\big) \;=\; \mathbb{E}_{\mathbf{a}^{\mathcal{B}} \sim \pi_s^{\mathcal{B}}(\cdot \,|\, h_t^{\mathcal{B}})}\Big[\, p\big(o_{t+1}, r_t \,\big|\, h_t, a_t, \mathbf{a}^{\mathcal{B}}\big)\,\Big]. \tag{4}$$

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

$$b_t(s) \;:=\; \Pr(s \mid h_t) \;\propto\; \Pr(s) \prod_{u=0}^{t-1} p\big(o_{u+1}, r_u \,\big|\, h_u, a_u;\, s\big). \tag{5}$$

For an ideal Bayesian agent, each timestep provides evidence about which partner population it faces, and the belief b_t can concentrate on the true scenario when scenarios are distinguishable from the focal history. 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

$$V^\star(h_t) \;=\; \max_{a_t} \mathbb{E}_{s \sim b_t}\Big[\, \mathbb{E}_{r_t,\, o_{t+1} \sim p(\cdot \,|\, h_t, a_t; s)}\big[\, r_t + \gamma\, V^\star(h_{t+1})\,\big]\,\Big]. \tag{6}$$

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 one of the difficulties that Settings 2 and 3 involve.

This formulation is an idealised reference point, not a requirement on learners: an agent that tracks b_t explicitly and one that reaches the same return through a learned generalist strategy are equally good under this view, and any partner-related reasoning in the latter is implicit in its recurrent state. Whether partner inference actually helps depends on whether best responses differ across scenarios and whether scenarios are distinguishable from history. We do not claim that Settings 2 or 3 can only be solved through partner inference, nor that a low J_{II} is due to its absence: multi-scenario (multi-task) learning, model estimation from finite data, extrapolation error, limited data support and conservatism can all contribute (Section 4.2), and neither setting isolates partner inference from these. When $\Pr(s) = P^{\mathrm{te}}$, $V^\star$ upper-bounds J_{II} (Eq. 3); in Setting 3 the training prior and generative models no longer match the test scenarios (Section 4.3), so this view serves only as a reference point.

C HYPERPARAMETERS

Tuning protocol. We purposefully use a modest hyperparameter tuning budget. For Evaluation Settings 1 and 2, hyperparameters were tuned on scenario 0 of each substrate only, and the resulting values are reused unchanged on every other scenario in that substrate. For Evaluation Setting 3, scenario 0 may lie in the test set of a split or in neither set. We therefore always tuned on the scenario with the lowest index in the split's *training* set, so that no held-out scenario informs hyperparameter selection. Commons Harvest indexes its scenarios separately within the Closed, Open and Partnership families; there we use the Closed family for this purpose, so that scenario 0 means Closed 0 and ties in index are broken in favour of Closed. It is now well-recognised that the online tuning budget for offline RL should be kept small and fixed (Formanek et al., 2024; Kurenkov & Kolesnikov, 2022), since online evaluation for hyperparameter selection is exactly the resource we cannot liberally spend in deployment. Future methods should be compared under a similarly small budget.

Shared settings. Every algorithm uses the same focal-agent architecture, optimiser and training schedule (Table 5). The remaining algorithm-specific hyperparameters are given in Table 6 for Evaluation Settings 1 and 2 and in Table 7 for Evaluation Setting 3.

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 for Evaluation Settings 1 and 2. Values were tuned on scenario 0 of each substrate and re-used unchanged for every other scenario in that 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	5.0

Table 7: Per-split hyperparameters for Evaluation Setting 3. Each split's hyperparameters were tuned on the lowest-index scenario in its training set ("Tuning scn."; scenario indices as in Table 4), and re-used for all scenarios of that split; splits whose tuning scenario is the one already used for Evaluation Settings 1 and 2 reuse the values of Table 6.

			IQL		BCQ	CQL
Substrate	**Split**	**Tuning scn.**	τ_{IQL}	β	τ_{BCQ}	α
Clean Up	U1	0	0.70	5.0	0.30	7.0
	U2	0	0.70	5.0	0.30	7.0
	U3	0	0.70	5.0	0.30	7.0
Coins	C1	0	0.80	2.0	0.30	3.0
	C2	1	0.80	3.0	0.40	5.0
	C3	1	0.80	3.0	0.40	5.0
	C4	1	0.80	3.0	0.40	5.0
Coop Mining	M1	0	0.80	4.0	0.30	5.0
	M2	0	0.80	4.0	0.30	5.0
	M3	0	0.80	4.0	0.30	5.0
Commons Harvest	H1	Closed 0	0.70	2.0	0.30	3.0
	H2	Closed 0	0.70	2.0	0.30	3.0
	H3	Closed 0	0.70	2.0	0.30	3.0
	H4	Partnership 0	0.80	3.0	0.40	5.0
Allelopathic Harvest	A1	1	0.70	5.0	0.50	10.0
	A2	3	0.70	3.0	0.40	5.0
	A3	3	0.70	3.0	0.40	5.0
	A4	0	0.70	3.0	0.40	5.0

D COMPUTE

Run count. Table 8 breaks down the total number of training runs in the benchmark by Evaluation Setting. Evaluation Setting 1 trains one policy per (scenario, algorithm, seed), Evaluation Setting 2 trains one policy per (substrate, algorithm, seed) on the substrate's full scenario union, and Evaluation Setting 3 trains one policy per (train/test split, algorithm, seed). With four algorithms (BC, BCQ, IQL, CQL) and three seeds throughout, this totals 840 training runs.

Table 8: Training run count by Evaluation Setting. Each configuration is trained with 4 algorithms $\times$ 3 seeds.

Evaluation Setting	Configurations	$\times$ Algorithms	$\times$ Seeds	Runs
Setting 1 (per scenario)	47 scenarios	4	3	564
Setting 2 (per-substrate union)	5 substrates	4	3	60
Setting 3 (train/test splits)	18 splits	4	3	216
Total				**840**

Run time. Wall-clock time per run is largely a function of the number of focal agents simulated in a scenario, since Melting Pot substrate stepping cost scales with the number of simulated agents. Coins has the fewest focal agents (one) and is the fastest substrate to train, at approximately 2 hours per run; Allelopathic Harvest has the most (around 14) and is the slowest, at approximately 9 hours per run. Table 9 gives the approximate per-run time for each substrate, the resulting total run count (Setting 1 + Setting 2 + Setting 3 combined), and the resulting GPU-hours.

Table 9: Approximate per-run time, total runs, and GPU-hours by substrate, summed across Evaluation Settings 1–3.

Substrate	Time / run	Runs	GPU-hours
Coins	$\sim$2h	144	288
Coop Mining	$\sim$3h	120	360
Clean Up	$\sim$4h	156	624
Commons Harvest	$\sim$5h	204	1020
Allelopathic Harvest	$\sim$9h	216	1944
Total		**840**	**4236**

Table 10 gives the same GPU-hour total broken down by Evaluation Setting instead of substrate. Evaluation Setting 1 dominates total compute, since it trains a separate policy per scenario rather than per substrate or split.

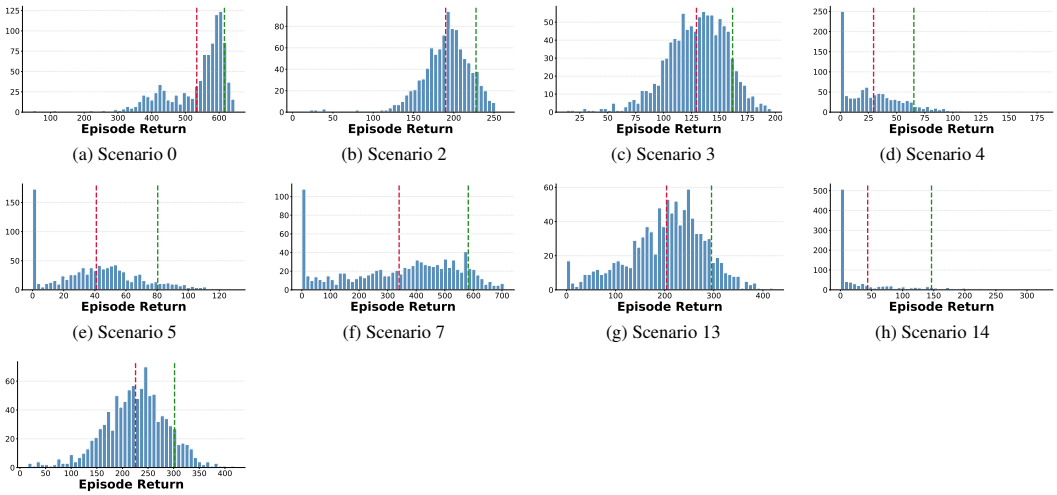
Table 10: Runs and GPU-hours by Evaluation Setting, summed across all five substrates.

Evaluation Setting	Runs	GPU-hours
Setting 1 (per scenario)	564	2940
Setting 2 (per-substrate union)	60	276
Setting 3 (train/test splits)	216	1020
Total	**840**	**4236**

Hardware and wall-clock time. All experiments were run on NVIDIA RTX 5090 GPUs, with four runs sharing a GPU concurrently. At 4236 total GPU-hours and 8 such GPUs (32 concurrent runs), producing the full benchmark takes approximately $4236/32 \approx 132$ hours, or about 5.5 days of wall-clock time.

E 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. (2026). 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.

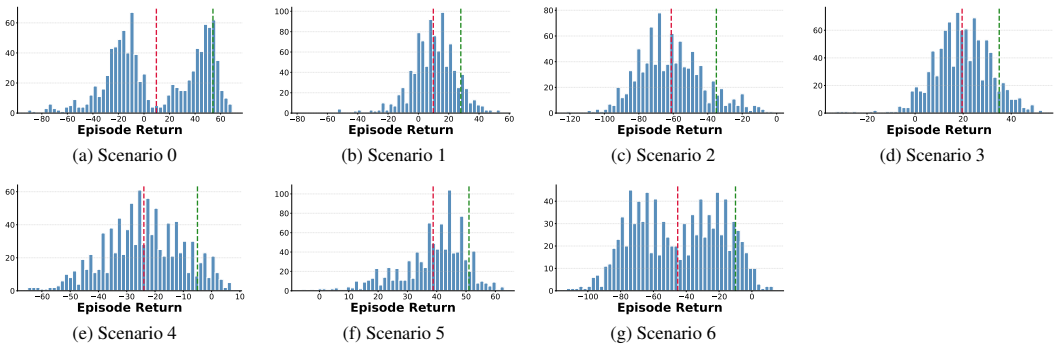


(a) Scenario 0 (b) Scenario 2 (c) Scenario 3 (d) Scenario 4

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

(i) Scenario 15

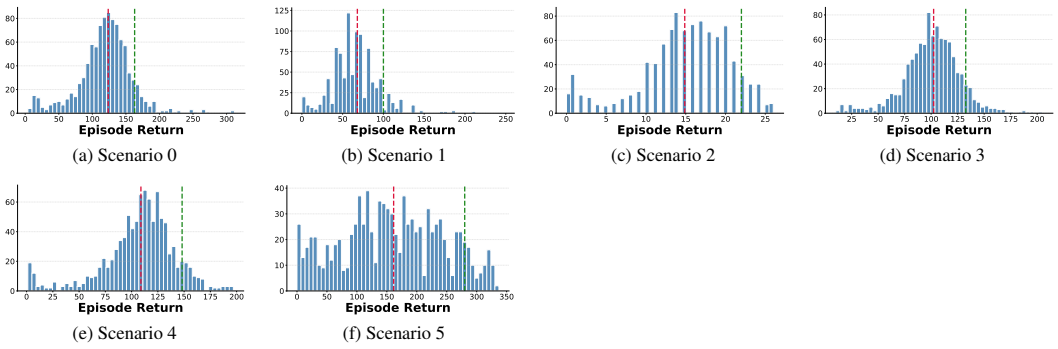
Figure 7: **Clean Up — per-scenario focal-return distributions.** Dashed red: scenario mean. Dashed green: 90th percentile.



(a) Scenario 0 (b) Scenario 1 (c) Scenario 2 (d) Scenario 3

(e) Scenario 4 (f) Scenario 5 (g) Scenario 6

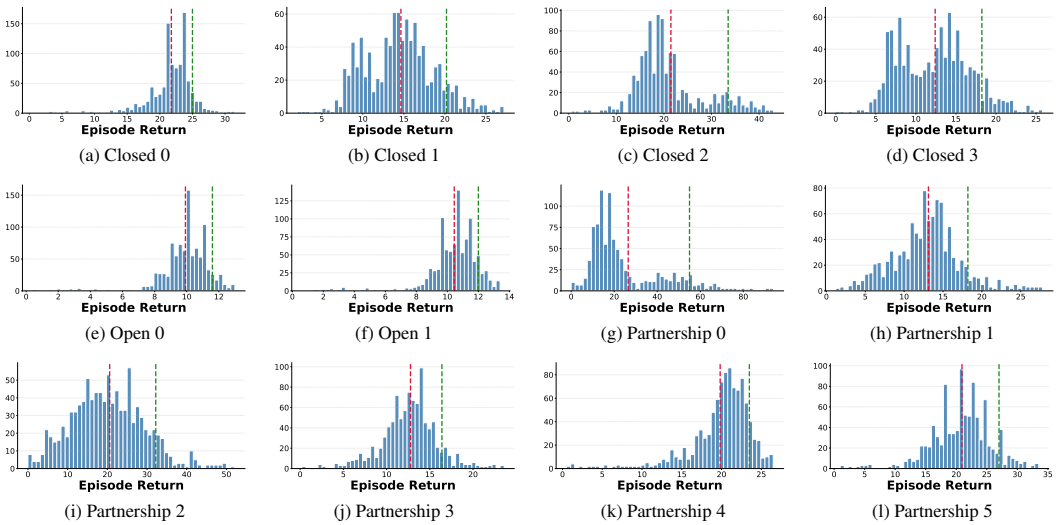
Figure 8: **Coins — per-scenario focal-return distributions.** Dashed red: scenario mean. Dashed green: 90th percentile.



(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.

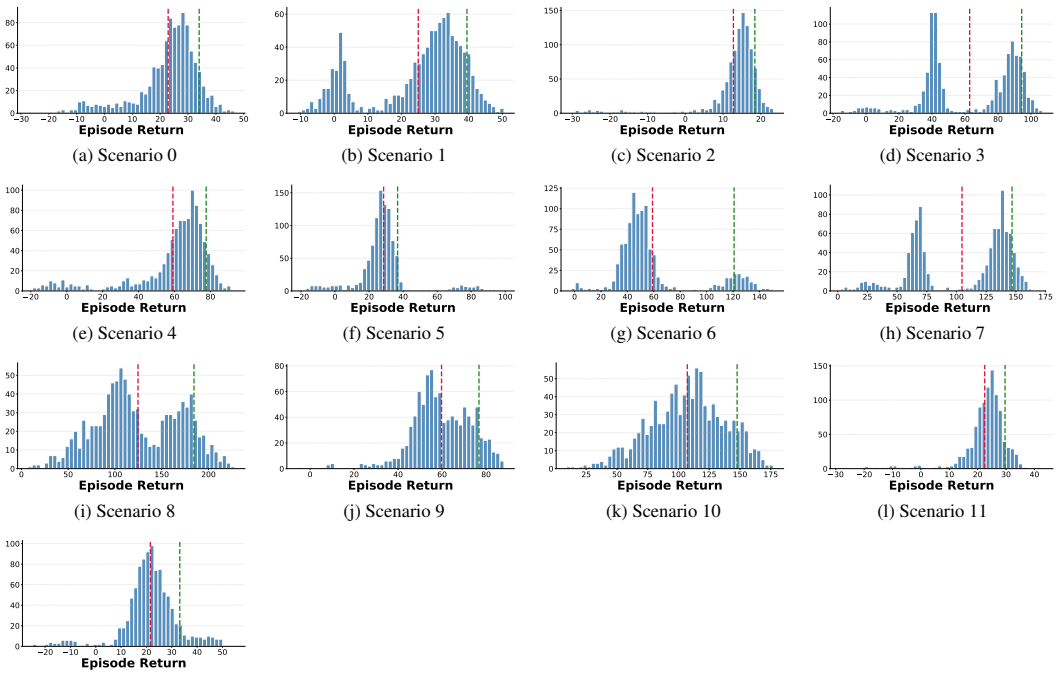


(a) Closed 0 (b) Closed 1 (c) Closed 2 (d) Closed 3

(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.

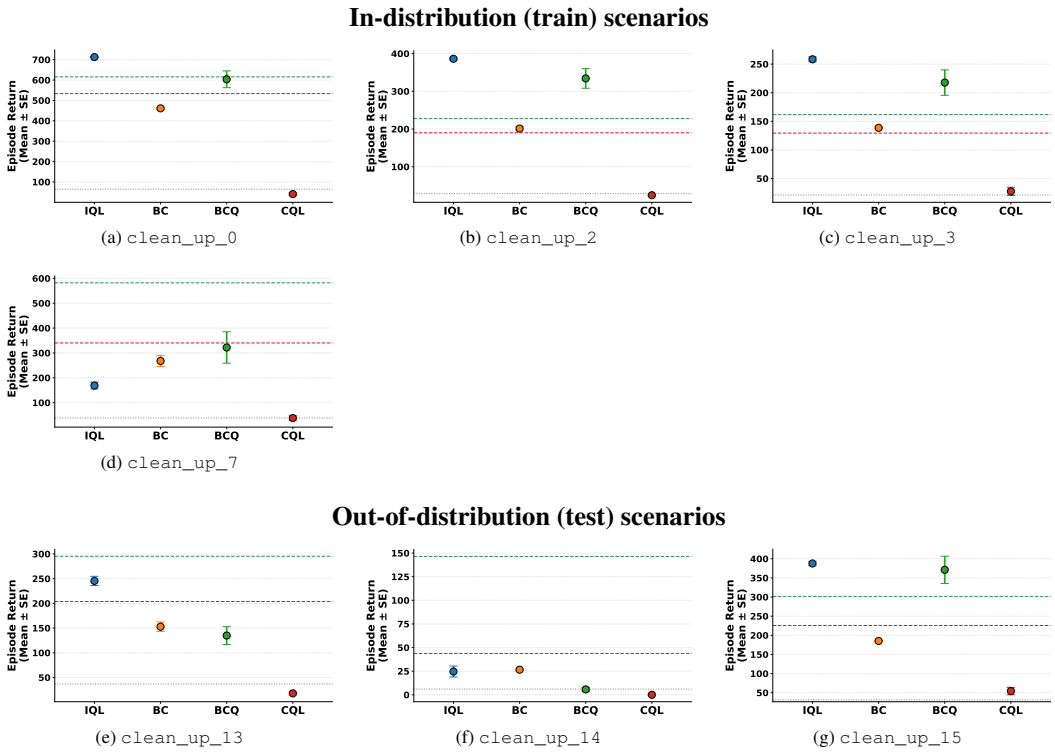
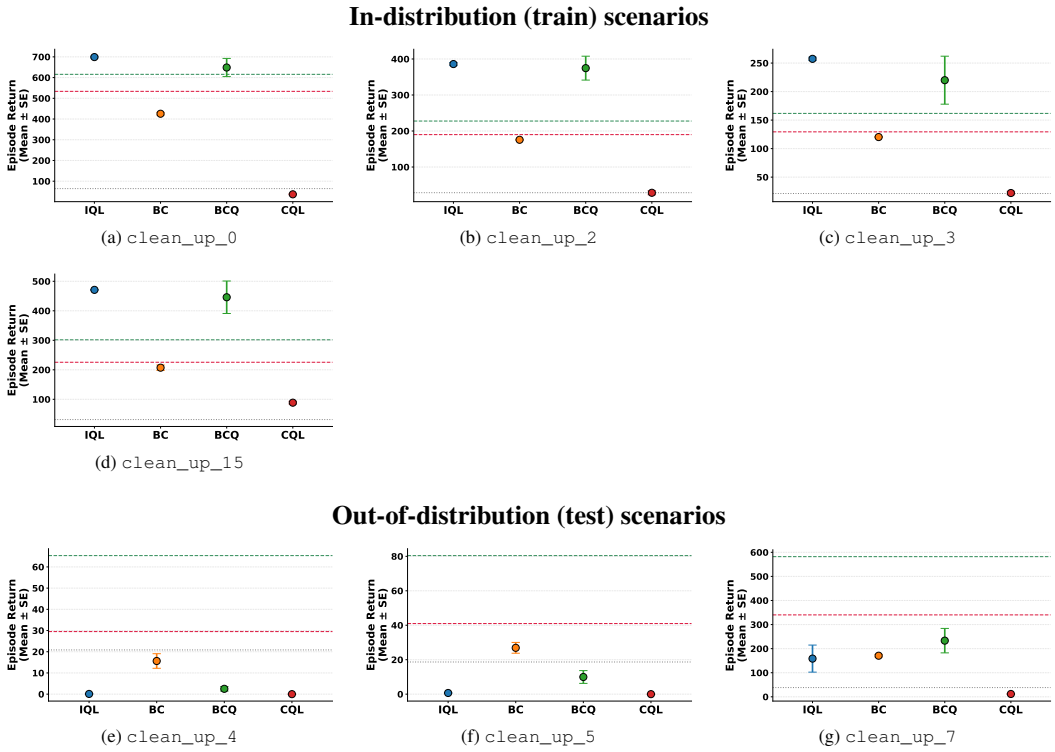


Figure 12: **Split U1 (Clean Up) — per-scenario Evaluation Setting 3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

F 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 (unnormalised) 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.



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

(d) `clean_up_15`

(e) `clean_up_4` (f) `clean_up_5` (g) `clean_up_7`

Figure 13: **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.

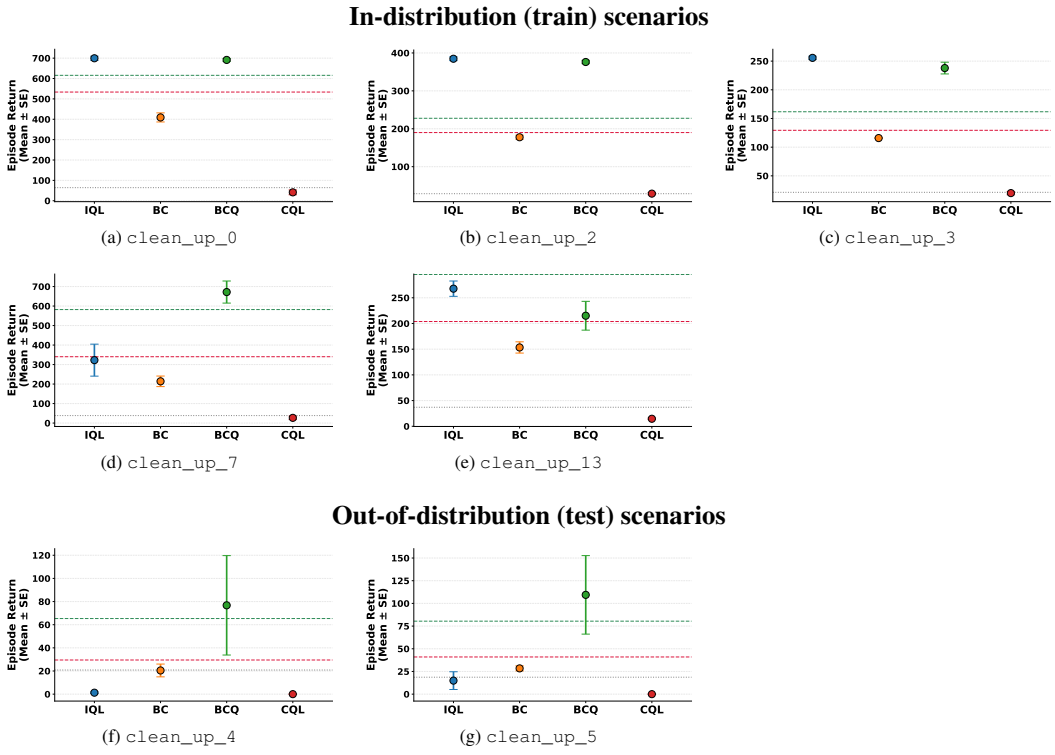
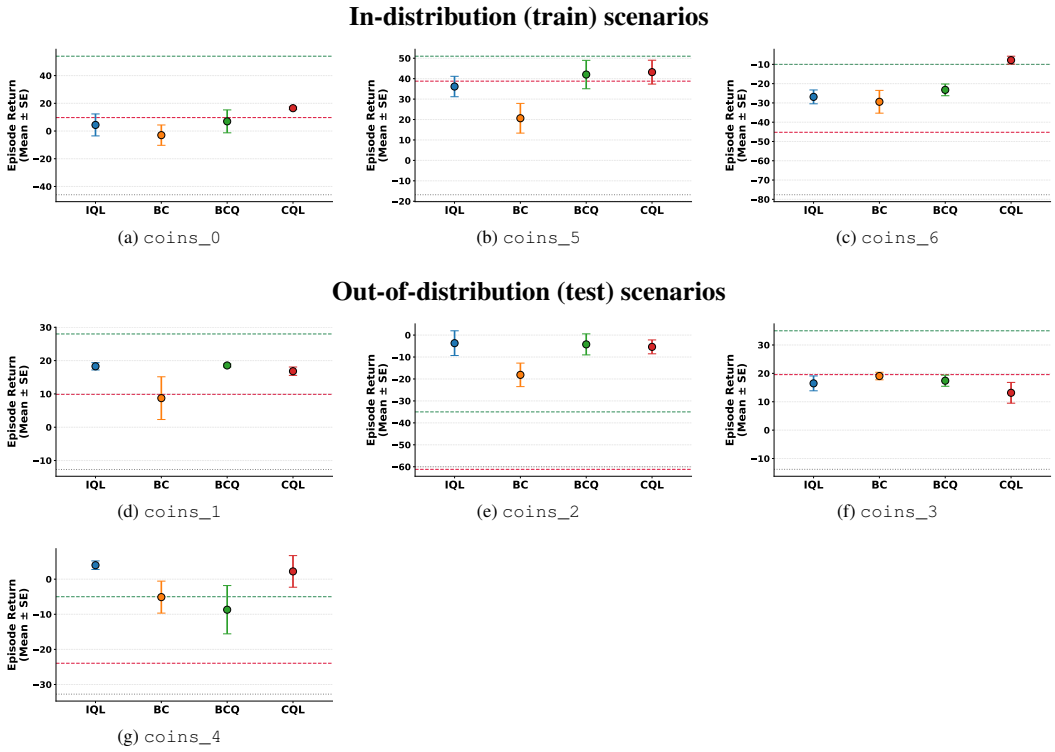


Figure 14: **Split U3 (Clean Up) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

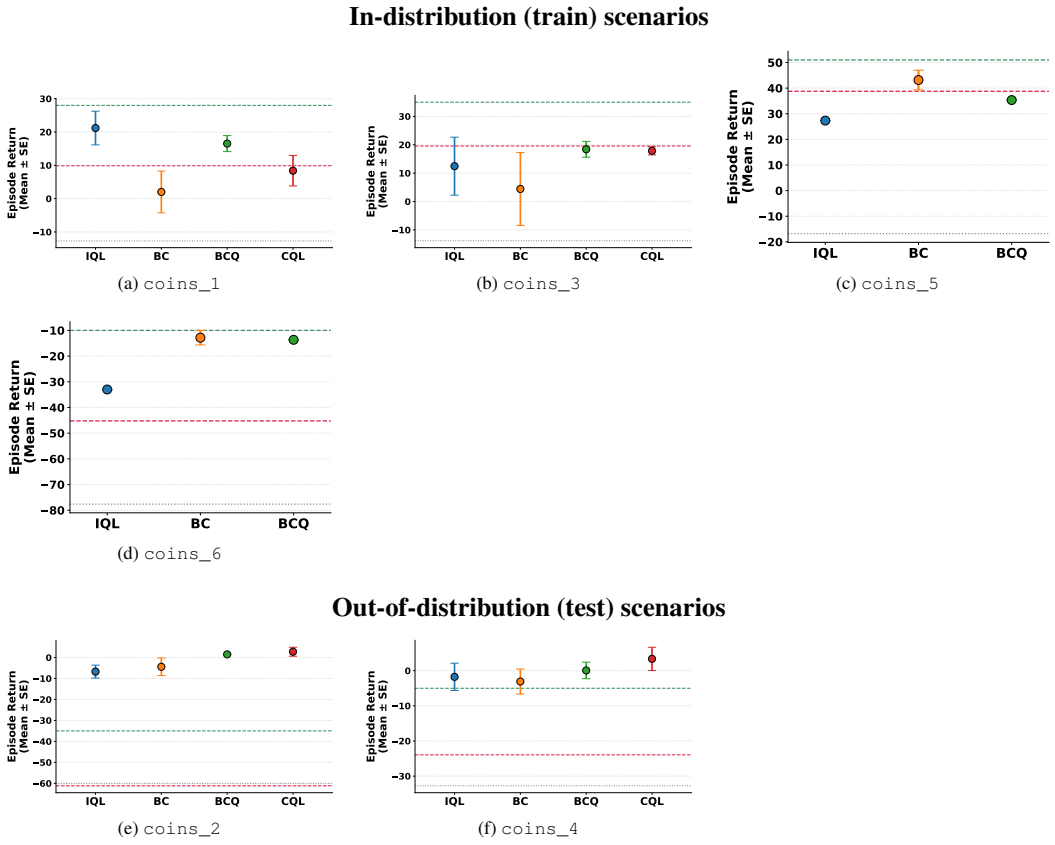


(a) coins_0 (b) coins_5 (c) coins_6

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

(g) coins_4

Figure 15: **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.

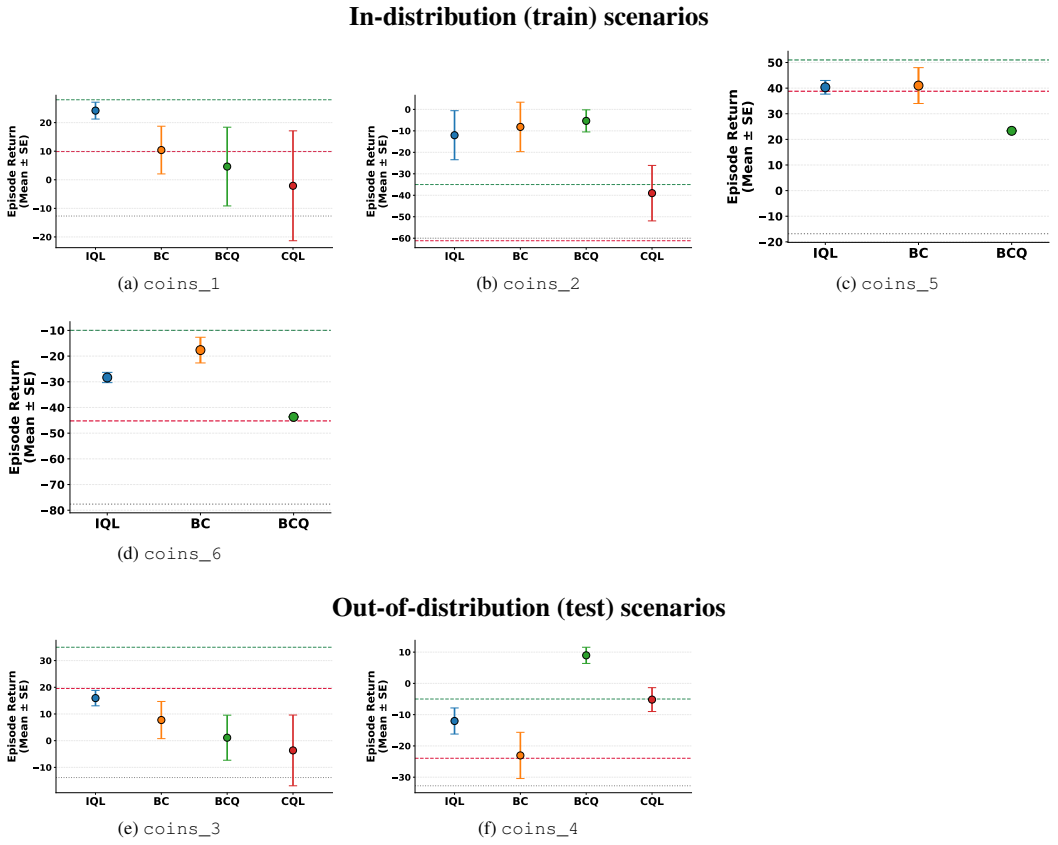


(a) coins_1 (b) coins_3 (c) coins_5

(d) coins_6

(e) coins_2 (f) coins_4

Figure 16: **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.

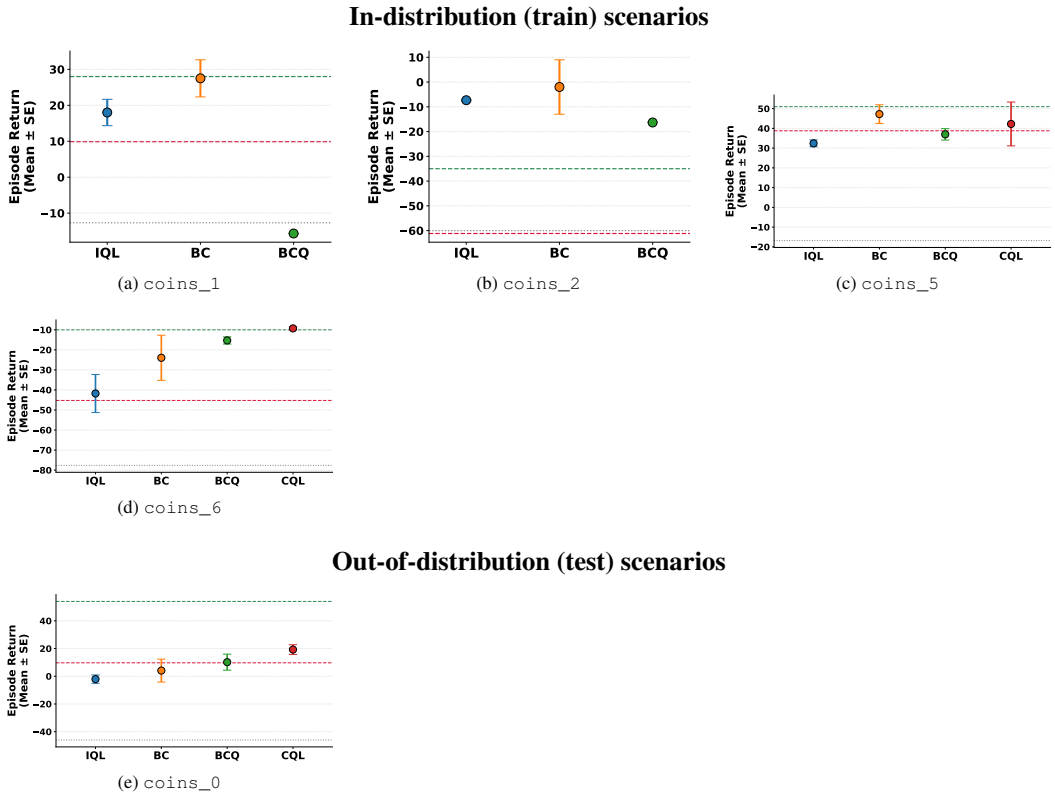


(a) coins_1 (b) coins_2 (c) coins_5

(d) coins_6

(e) coins_3 (f) coins_4

Figure 17: **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.

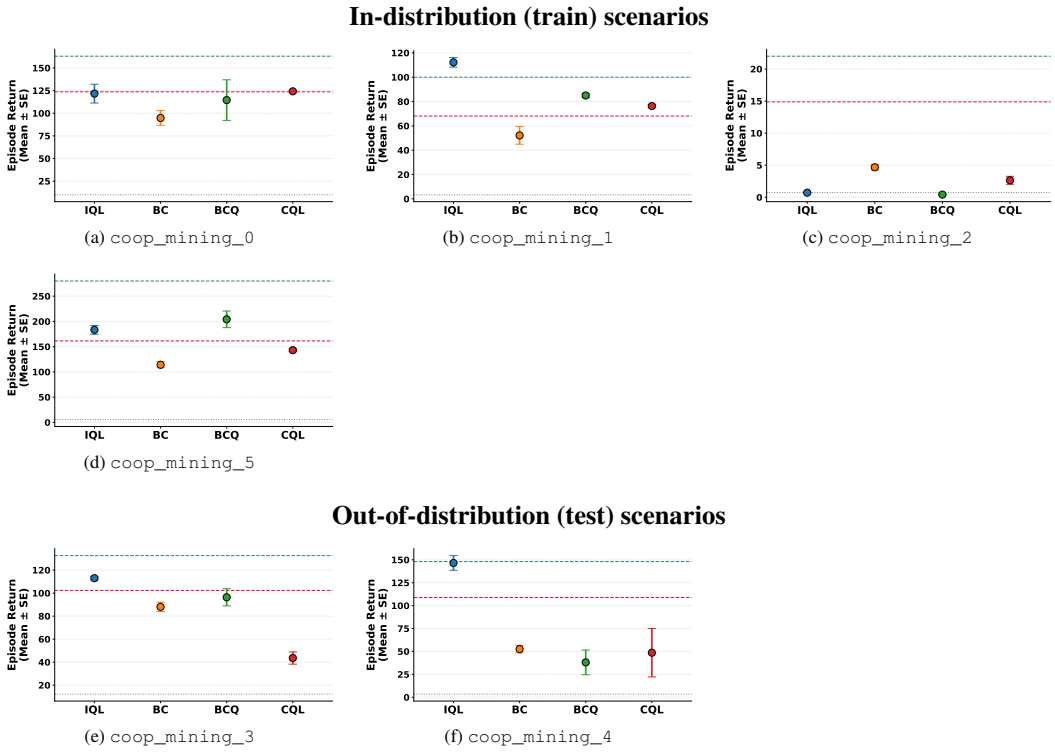


(a) coins_1 (b) coins_2 (c) coins_5

(d) coins_6

(e) coins_0

Figure 18: **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.

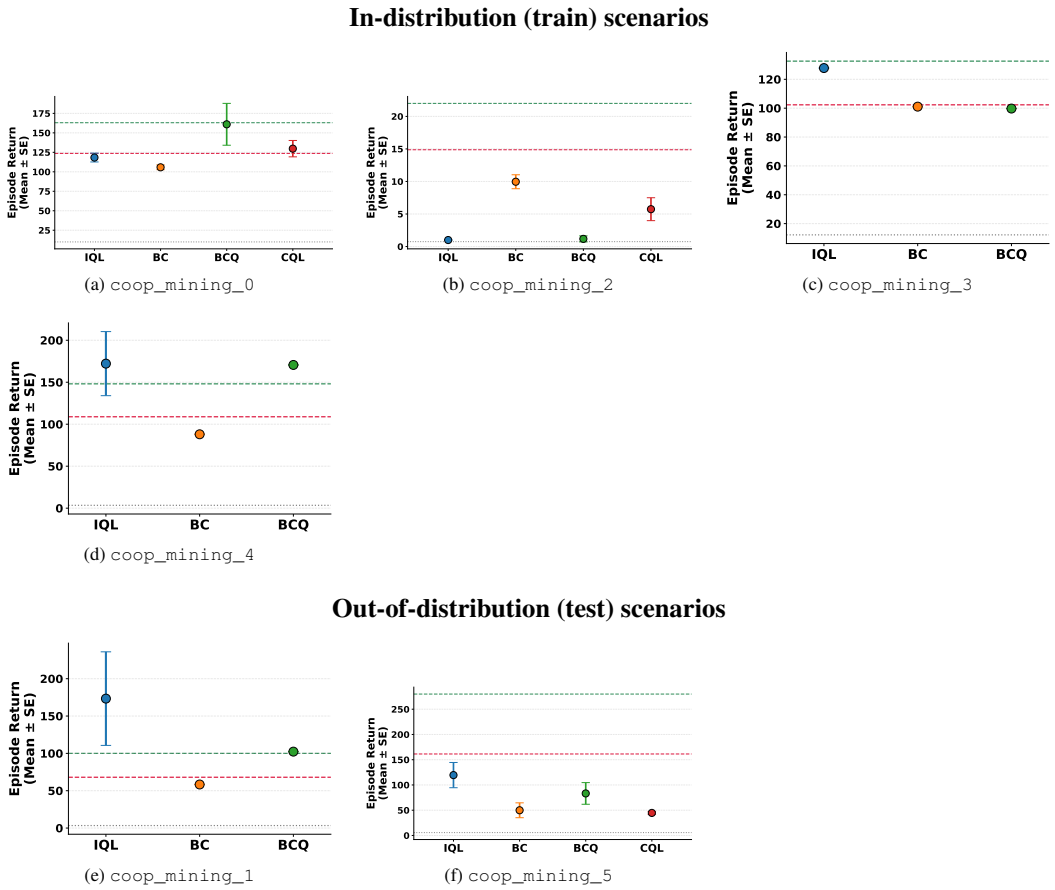


(a) coop_mining_0 (b) coop_mining_1 (c) coop_mining_2

(d) coop_mining_5

(e) coop_mining_3 (f) coop_mining_4

Figure 19: **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.

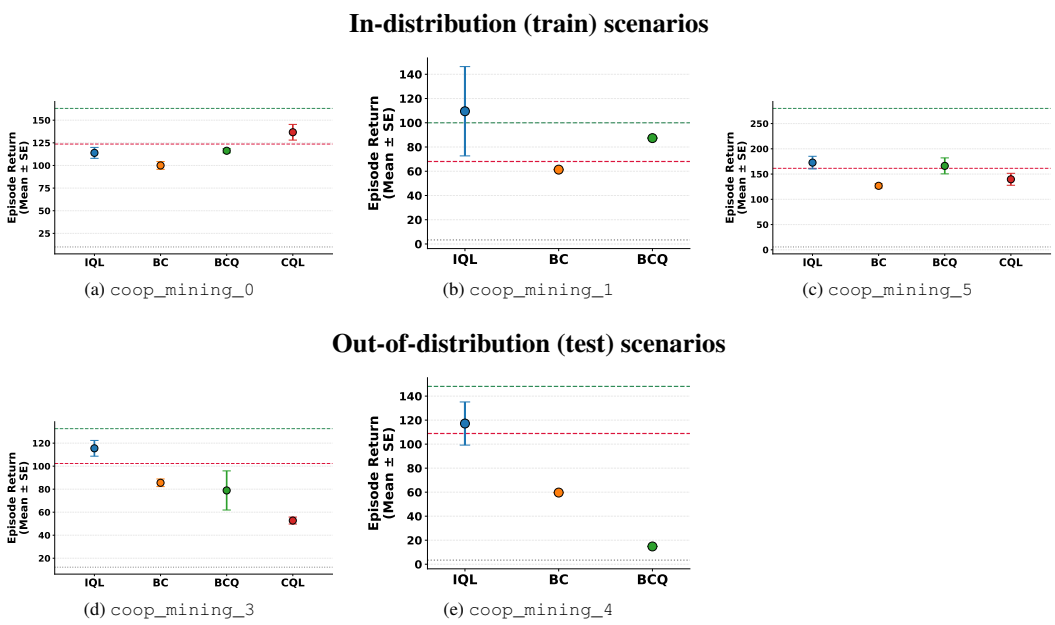


(a) coop_mining_0 (b) coop_mining_2 (c) coop_mining_3

(d) coop_mining_4

(e) coop_mining_1 (f) coop_mining_5

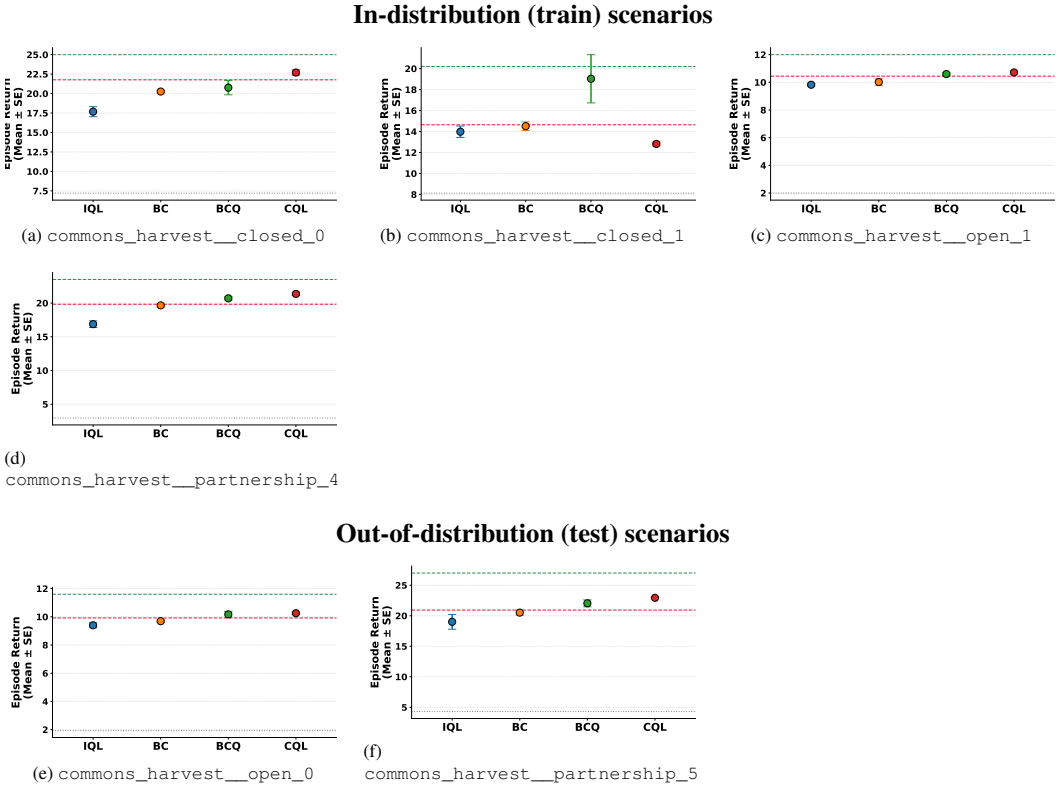
Figure 20: **Split M2 (Coop Mining) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.



(a) coop_mining_0 (b) coop_mining_1 (c) coop_mining_5

(d) coop_mining_3 (e) coop_mining_4

Figure 21: **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.



(a) commons_harvest__closed_0 (b) commons_harvest__closed_1 (c) commons_harvest__open_1

(d) commons_harvest__partnership_4

(e) commons_harvest__open_0 (f) commons_harvest__partnership_5

Figure 22: **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.

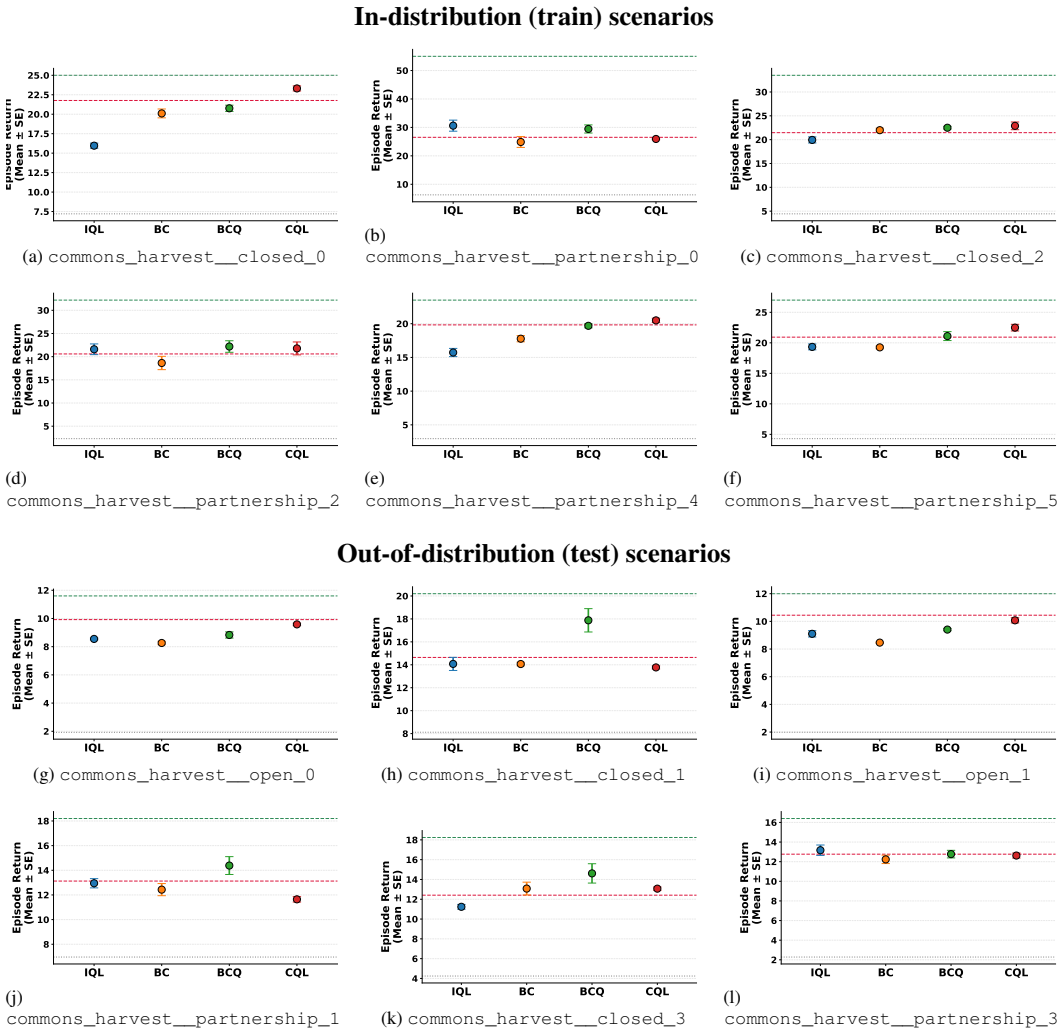
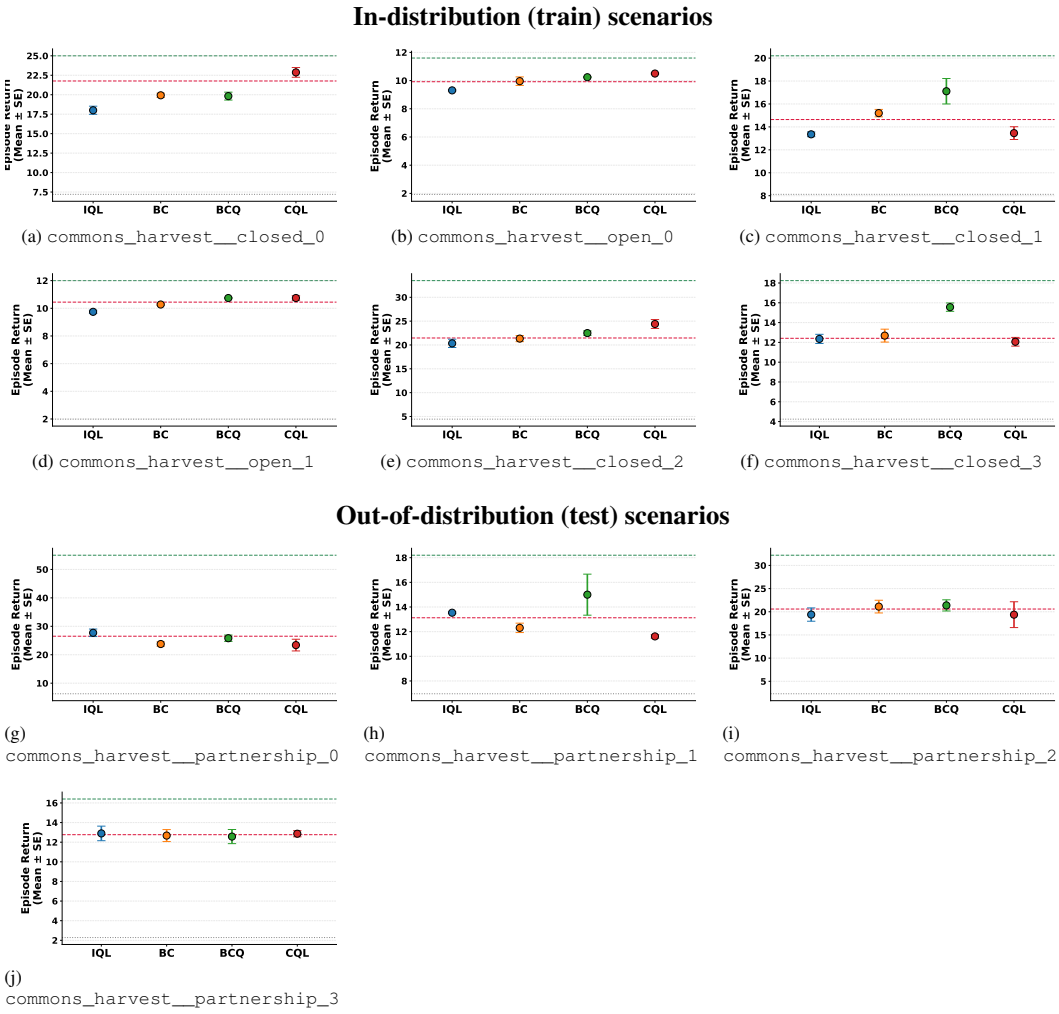


Figure 23: **Split H2 (Commons Harvest) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.



(a) commons_harvest__closed_0 (b) commons_harvest__open_0 (c) commons_harvest__closed_1

(d) commons_harvest__open_1 (e) commons_harvest__closed_2 (f) commons_harvest__closed_3

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

(j) commons_harvest__partnership_3

Figure 24: **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.

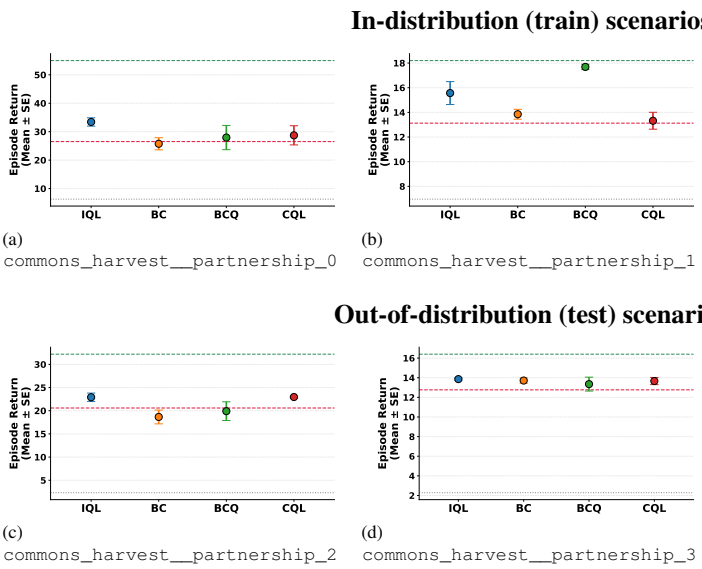


Figure 25: **Split H4 (Commons Harvest) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

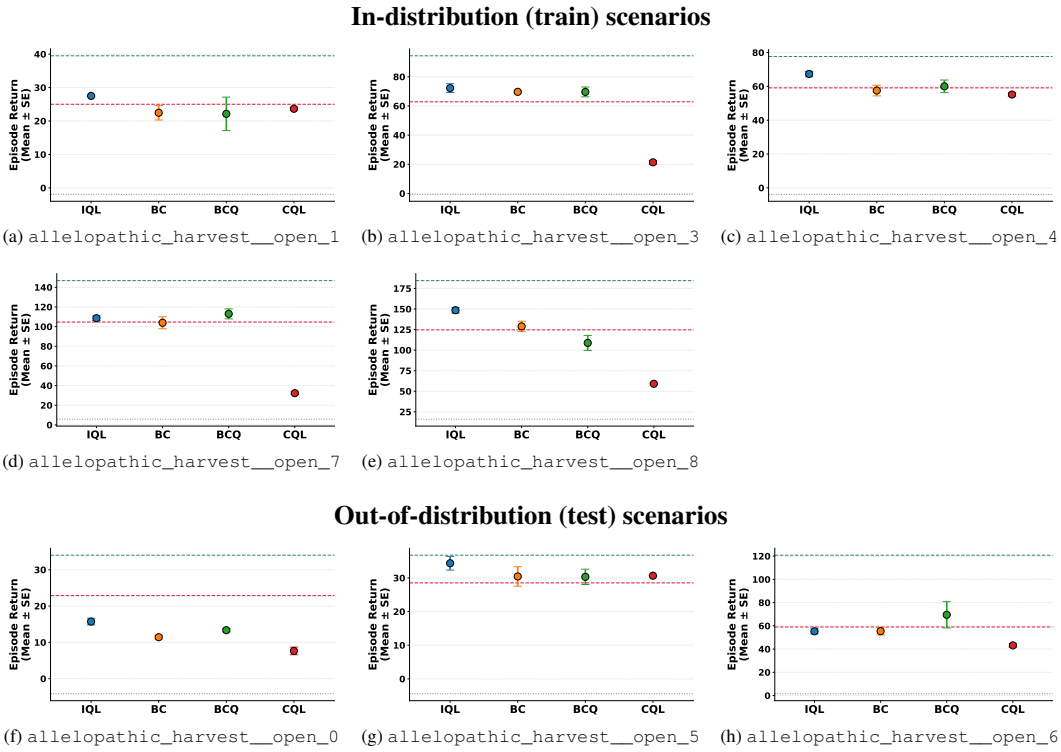


Figure 26: **Split A1 (Allelopathic Harvest) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

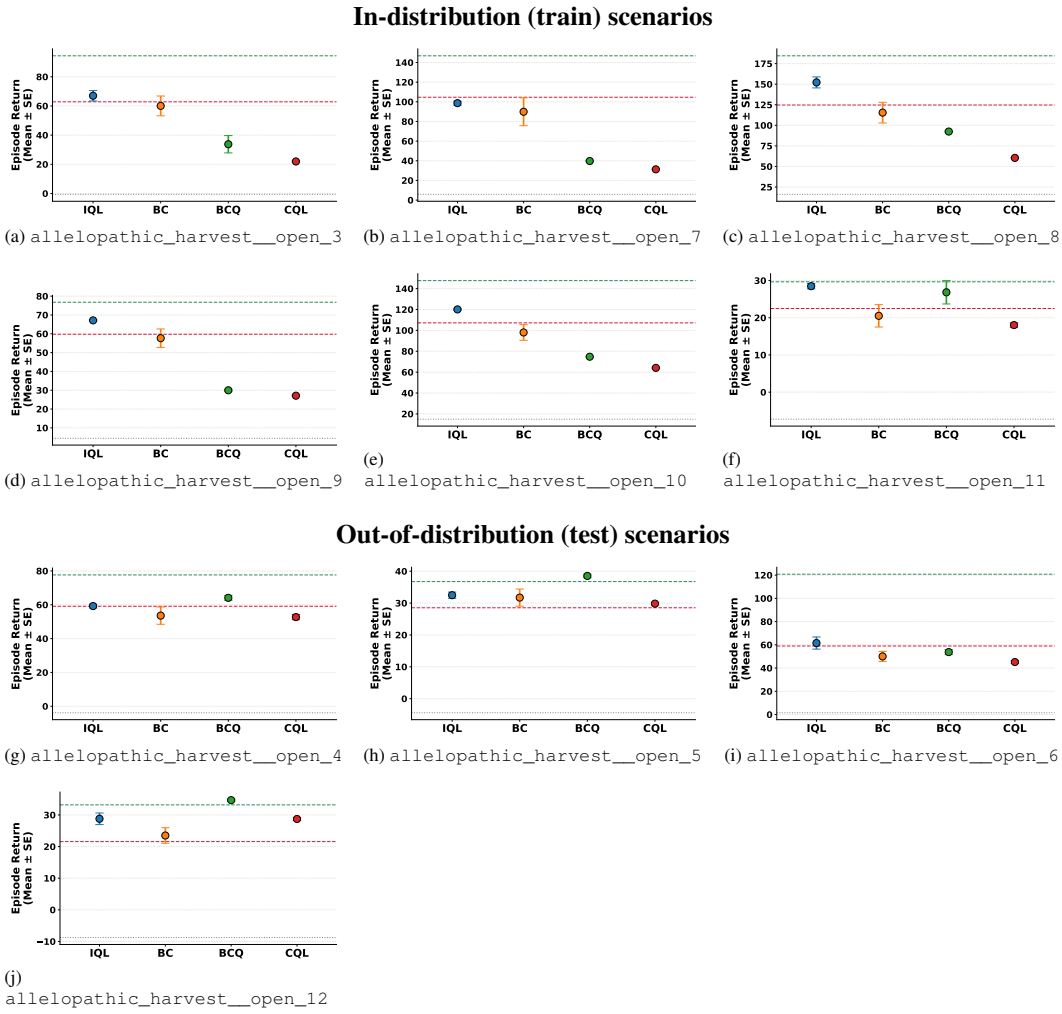


Figure 27: **Split A2 (Allelopathic Harvest) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.

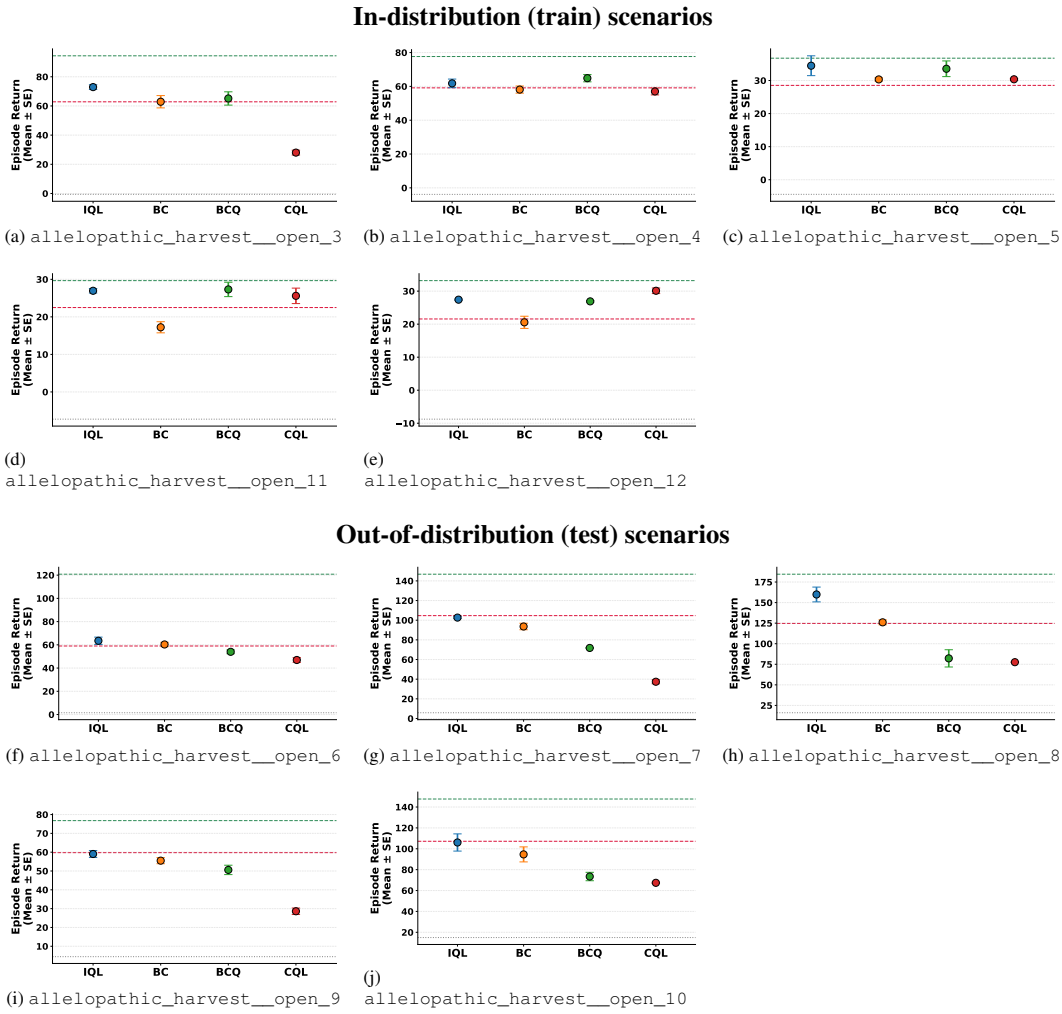
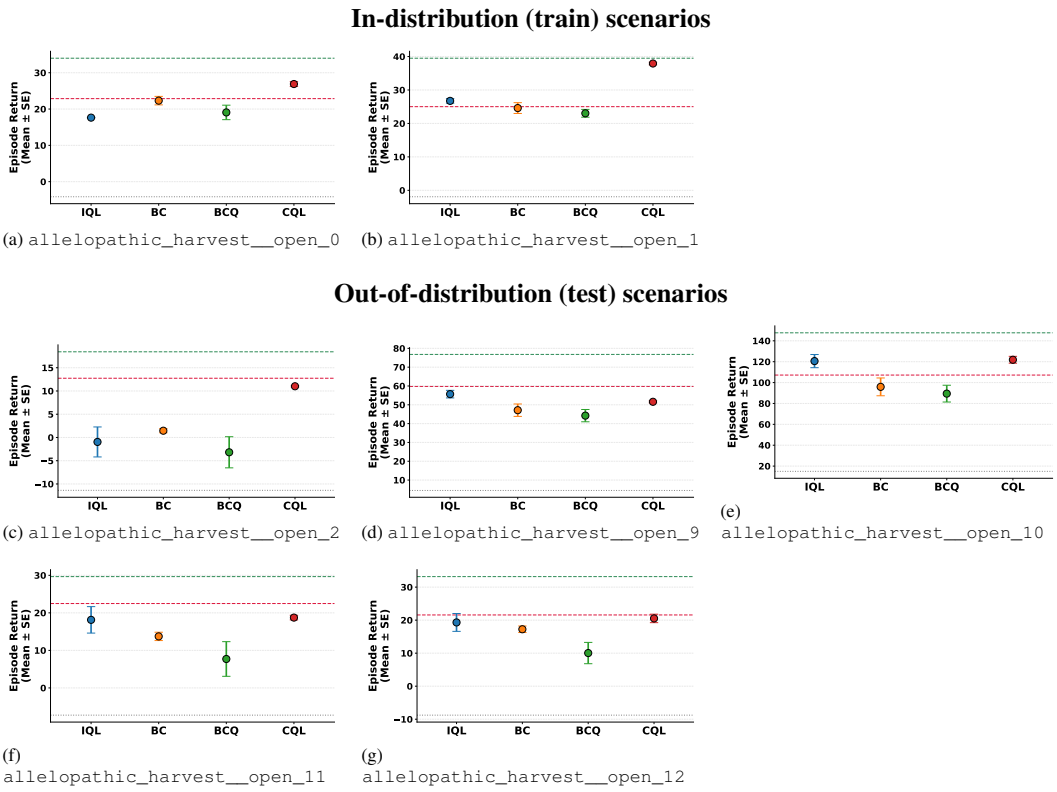


Figure 28: **Split A3 (Allelopathic Harvest) — per-scenario Evaluation Setting-3 eval returns.** Each panel shows one scenario's final eval return per algorithm (marker = mean across seeds, error bar = SE). Reference lines: dataset mean (red dashed), p_{90} best-in-dataset ceiling (green dashed), random-policy baseline (grey dotted). Y-axes are scaled per panel.



(a) `allelopathic_harvest__open_0` (b) `allelopathic_harvest__open_1`

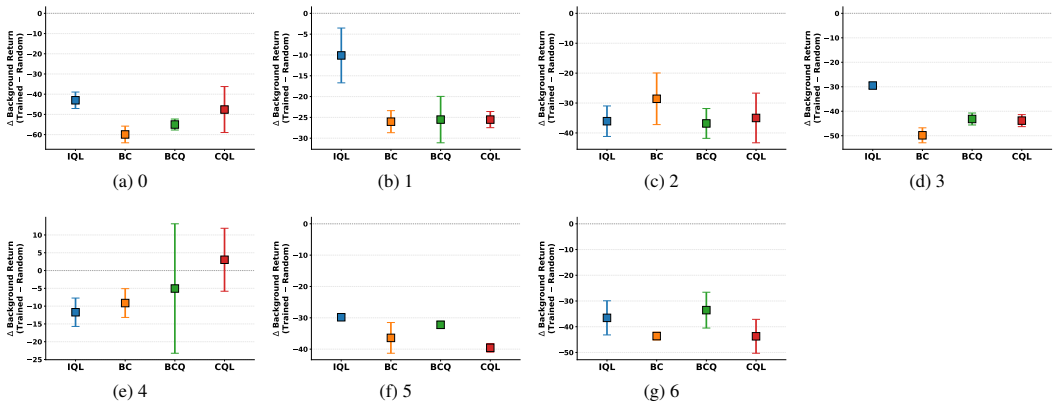
(c) `allelopathic_harvest__open_2` (d) `allelopathic_harvest__open_9` (e) `allelopathic_harvest__open_10`

(f) `allelopathic_harvest__open_11` (g) `allelopathic_harvest__open_12`

Figure 29: **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.

G PER-SCENARIO BACKGROUND-AGENT RETURN CHANGE

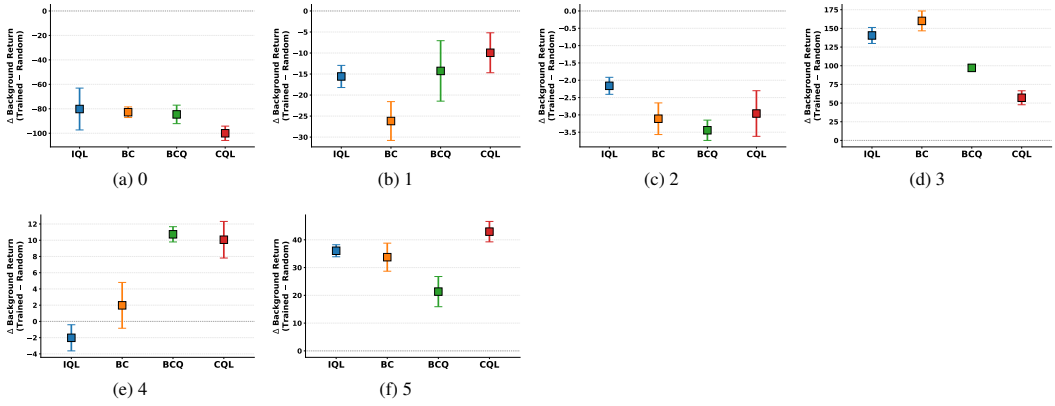
For each Evaluation 1 scenario we report the per-algorithm change in mean background-agent episode return induced by the trained focal policy, relative to a uniform-random focal policy ($\Delta = R_{\text{trained}} - R_{\text{random}}$). Background returns are read from the in-training eval summary (`eval/in_dist/<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.



(a) 0 (b) 1 (c) 2 (d) 3

(e) 4 (f) 5 (g) 6

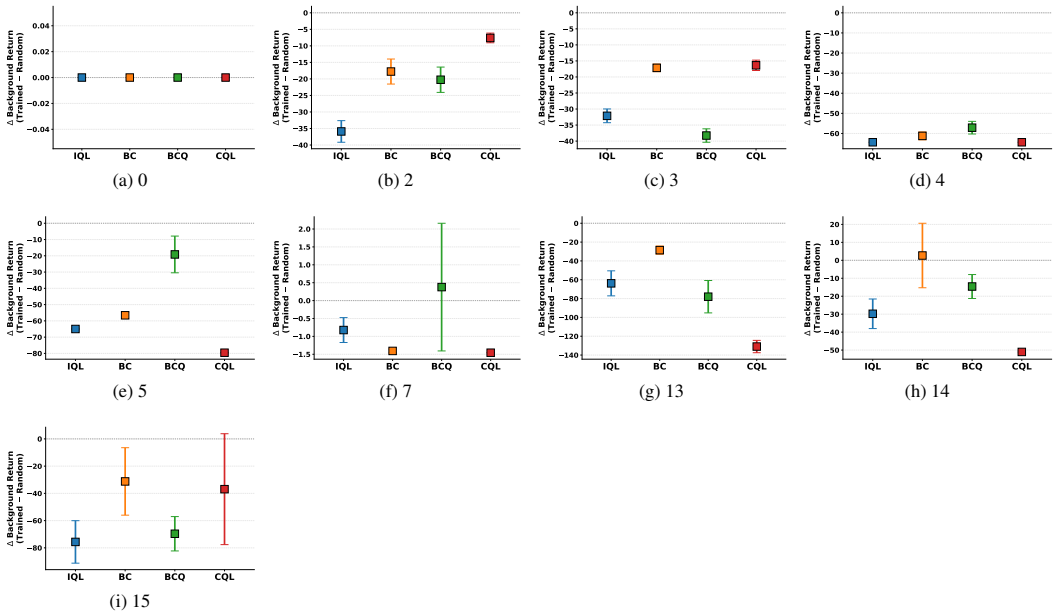
Figure 30: **Coins — background-agent return change.** For each scenario in Coins, one marker per algorithm showing the change in mean background-agent episode return induced by the trained focal policy, relative to a uniform-random focal policy ($\Delta = R_{\text{trained}} - R_{\text{random}}$). Background 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.



(a) 0 (b) 1 (c) 2 (d) 3

(e) 4 (f) 5

Figure 31: **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.

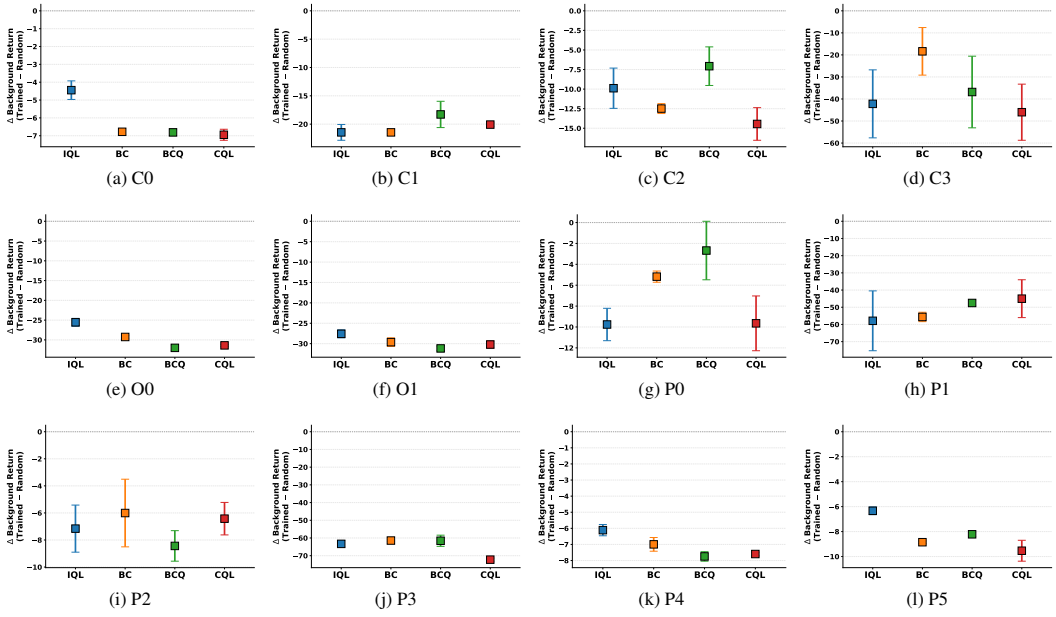


(a) 0 (b) 2 (c) 3 (d) 4

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

(i) 15

Figure 32: **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.

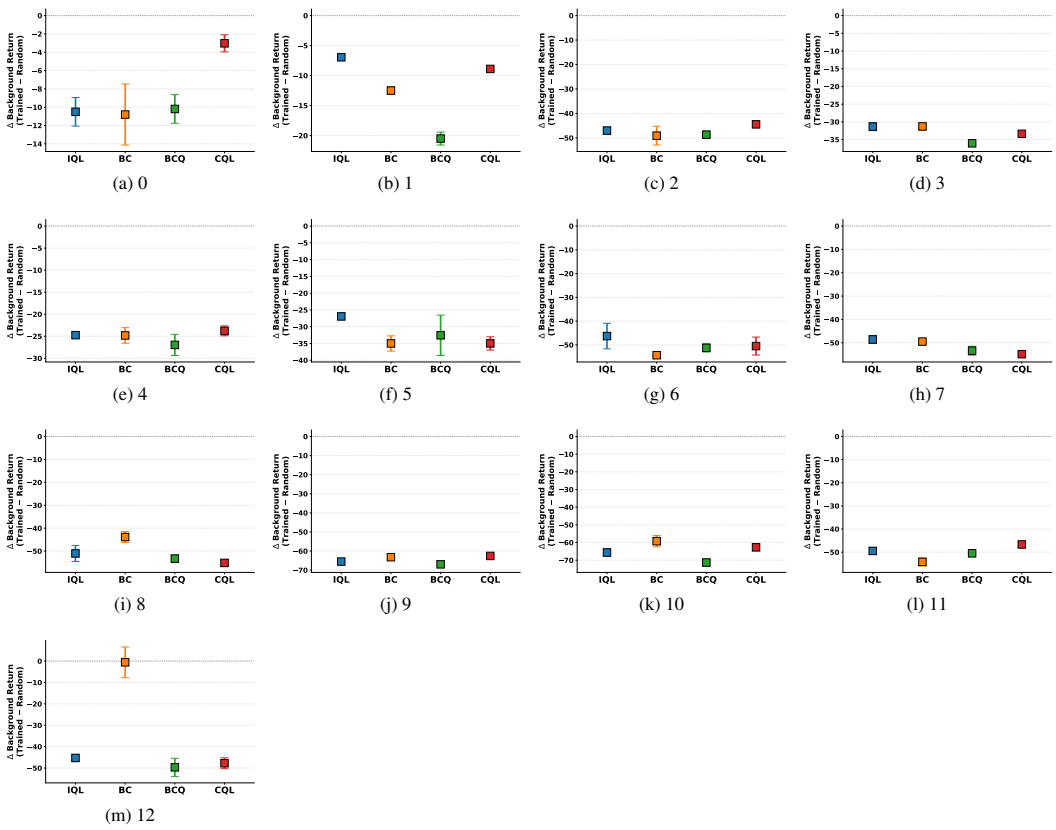


(a) C0 (b) C1 (c) C2 (d) C3

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

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

Figure 33: **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.



(a) 0 (b) 1 (c) 2 (d) 3

(e) 4 (f) 5 (g) 6 (h) 7

(i) 8 (j) 9 (k) 10 (l) 11

(m) 12

Figure 34: **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.

H CROSS-SUBSTRATE AGGREGATE SCORES

Table 11 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 11: **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 (Agarwal et al., 2021). Higher is better for the first three rows; optimality gap (↓). Baseline scores are a floor for future methods, not a controlled comparison.

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]

I LLM USAGE

In accordance with the ICLR policy on AI assistance, we disclose that large language model (LLM) assistants were used in preparing this paper in two ways. First, to aid and polish writing: condensing and rephrasing passages the authors had drafted, correcting grammar, and formatting LaTeX tables and figures. Second, for retrieval and discovery: suggesting candidate related work and pointers into the literature, each of which the authors read and verified before citing. Beyond the writing, we also used AI coding assistants while developing the benchmark code, the dataset pipeline and the experiment configurations, and while producing the figures and tables in this paper. The research ideas, benchmark design, experiments and analysis are the authors' own, and all code, text, citations and reported numbers were reviewed by the authors, who take full responsibility for the content of this paper.