

Melting Pot 2.0

John P. Agapiou^{*,1}, Alexander Sasha Vezhnevets^{*,1}, Edgar A. Duéñez-Guzmán^{*,1}, Jayd Matyas¹, Yiran Mao¹, Peter Sunehag¹, Raphael Köster¹, Udari Madhushani¹, Kavya Kopparapu¹, Ramona Comanescu¹, DJ Strouse¹, Michael B. Johanson¹, Sukhdeep Singh¹, Julia Haas¹, Igor Mordatch², Dean Mobbs³ and Joel Z. Leibo^{*,1}

^{*}Equal contributions, ¹DeepMind, ²Google, ³California Institute of Technology

Multi-agent artificial intelligence research promises a path to develop intelligent technologies that are more human-like and more human-compatible than those produced by “solipsistic” approaches, which do not consider interactions between agents. Melting Pot is a research tool developed to facilitate work on multi-agent artificial intelligence, and provides an evaluation protocol that measures generalization to novel social partners in a set of canonical test scenarios. Each scenario pairs a physical environment (a “substrate”) with a reference set of co-players (a “background population”), to create a social situation with substantial interdependence between the individuals involved. For instance, some scenarios were inspired by institutional-economics-based accounts of natural resource management and public-good-provision dilemmas. Others were inspired by considerations from evolutionary biology, game theory, and artificial life. Melting Pot aims to cover a maximally diverse set of interdependencies and incentives. It includes the commonly-studied extreme cases of perfectly-competitive (zero-sum) motivations and perfectly-cooperative (shared-reward) motivations, but does not stop with them. As in real-life, a clear majority of scenarios in Melting Pot have mixed incentives. They are neither purely competitive nor purely cooperative and thus demand successful agents be able to navigate the resulting ambiguity. Here we describe Melting Pot 2.0, which revises and expands on Melting Pot. We also introduce support for scenarios with asymmetric roles, and explain how to integrate them into the evaluation protocol. This report also contains: (1) details of all substrates and scenarios; (2) a complete description of all baseline algorithms and results. Our intention is for it to serve as a reference for researchers using Melting Pot 2.0.

Keywords: Multi-Agent Reinforcement Learning, Zero-Shot Generalization, AI Evaluation

| Contents | |
|-------------------------------------------------------------------------|-----------|
| Introduction | 2 |
| Training and testing | 5 |
| Roles | 5 |
| Substrate factories | 5 |
| Scenarios | 5 |
| Secondary evaluation metrics | 5 |
| Benchmark experiments | 6 |
| Agent architectures | 6 |
| Computational setup | 8 |
| Agents hyper-parameters | 8 |
| Normalising results | 9 |
| Results summary | 11 |
| Train & test scores as a function of training time | 12 |
| Discussion | 12 |
| Substrates and Scenarios | 12 |
| Common properties shared by all substrates | 12 |
| Allelopathic Harvest | 13 |
| Boat Race: Eight Races | 14 |
| Chemistry * | 15 |
| Chemistry: Three Metabolic Cycles | 16 |
| Chemistry: Three Metabolic Cycles With Plentiful Distractors | 17 |

| | | | |
|---------------------------------------------------------------|----|------------------------------------------------------------------|----|
| Chemistry: Two Metabolic Cycles | 17 | Bach or Stravinsky in the matrix: Repeated | 45 |
| Chemistry: Two Metabolic Cycles With Distractors | 18 | Chicken in the matrix: Arena | 46 |
| Clean Up | 19 | Chicken in the matrix: Repeated | 47 |
| Coins | 21 | Prisoners Dilemma in the matrix: Arena | 48 |
| Collaborative Cooking | 22 | Prisoners Dilemma in the matrix: Repeated | 49 |
| Collaborative Cooking: Asymmetric | 22 | Pure Coordination in the matrix: Arena | 50 |
| Collaborative Cooking: Circuit | 23 | Pure Coordination in the matrix: Repeated | 51 |
| Collaborative Cooking: Cramped | 23 | Rationalizable Coordination in the matrix: Arena | 52 |
| Collaborative Cooking: Crowded | 23 | Rationalizable Coordination in the matrix: Repeated | 53 |
| Collaborative Cooking: Figure Eight | 24 | Running With Scissors in the ma- trix: Arena | 53 |
| Collaborative Cooking: Forced | 24 | Running With Scissors in the ma- trix: One Shot | 54 |
| Collaborative Cooking: Ring | 25 | Running With Scissors in the ma- trix: Repeated | 55 |
| Commons Harvest * | 25 | Stag Hunt in the matrix: Arena | 56 |
| Commons Harvest: Closed | 25 | Stag Hunt in the matrix: Repeated | 57 |
| Commons Harvest: Open | 26 | | |
| Commons Harvest: Partnership | 27 | | |
| Coop Mining | 28 | | |
| Daycare | 29 | | |
| Externality Mushrooms: Dense | 30 | | |
| Factory of the Commons: Either Or | 31 | | |
| Fruit Market: Concentric Rivers | 32 | | |
| Gift Refinements | 33 | | |
| Hidden Agenda | 34 | | |
| Paintball * | 35 | | |
| Paintball: Capture the Flag | 36 | | |
| Paintball: King of the Hill | 36 | | |
| Predator-Prey * | 37 | | |
| Predator Prey: Alley Hunt | 37 | | |
| Predator Prey: Open | 38 | | |
| Predator Prey: Orchard | 39 | | |
| Predator Prey: Random Forest | 40 | | |
| Territory * | 40 | | |
| Territory: Inside Out | 40 | | |
| Territory: Open | 41 | | |
| Territory: Rooms | 42 | | |
| * in the Matrix | 43 | | |
| Bach or Stravinsky in the matrix: Arena | 44 | | |
| | | Reference learning curves | 58 |

Introduction

AI research is often divided into one branch that aims to improve the fundamental capabilities of AI systems, and another branch that aims to ensure AI is used in a way that benefits humanity. Research in the fundamental capabilities branch is typically concerned with solipsistic metrics such as performance in perception and cognition challenges (e.g. [Crosby et al. \(2019\)](#); [Koehn and Knowles \(2017\)](#); [Ott et al. \(2022\)](#); [Russakovsky et al. \(2015\)](#)). Research in the second branch is typically concerned with outcomes arising from introducing AI technology in real human societies like perpetuating unfair biases, diminishing privacy, promoting misinformation, and exacerbating inequality (e.g. [Acemoglu \(2021\)](#); [Mehrabi et al. \(2021\)](#); [Obermeyer et al. \(2019\)](#); [Rahwan et al. \(2019\)](#); [Weidinger et al. \(2021\)](#)). This separation can lead to researchers in the capabilities

branch to construct and deploy systems without enough consideration of their likely social consequences, and engaging with researchers in the beneficence branch only after the fact. This is not ideal. We think there should be a way to make fundamental capabilities research less solipsistic.

Of course, it will always be desirable to measure the social consequences of AI deployment in the real world. Nevertheless, we may still benefit from developing challenges that are less solipsistic, while still being usable by fundamental-capabilities researchers engaged in the earlier stages of AI technology development. At core, AI provides ways of simulating agents. So we should be able to use it in a self-reflective way that constructs models of societies where multiple agents interact, and social consequences can be simulated. It may work similarly to agent-based modeling approaches in economics (e.g. [Tsfatjion \(2021\)](#)), but here using individuals modeled by learning agents. By adopting such an approach, early stage fundamental capabilities research need not be entirely blind to questions concerning how technology can be deployed in a socially optimal way. This is the human compatibility purpose of Melting Pot.

Melting Pot was also constructed to facilitate research on human-like AI, adopting the stance that it is our ultra-sociality that makes human intelligence unique. Along most of the traditionally studied domains of intelligence we do not outshine other mammals: our perception, memory, attention, planning, and decision-making skills are nothing special for a mammal of our size ([Henrich and Muthukrishna, 2021](#)). Rather we distinguish ourselves primarily along *social-cognitive* dimensions. For instance, we improvisationally achieve cooperation through means such as persuasion and negotiation, reciprocity and reputational concern, alliances and incentive alignment, imitation and teaching, leadership and responsibility, commitments and promises, trust and monitoring, sharing and division of labor, social norms and institutions, and other social-cognitive capacities, representations, motivations, and mechanisms.

We propose an approach to building advanced AI systems based on reverse-engineering human

intelligence. This reverse engineering approach is common in AI, especially with people who work on the “classic” cognitive abilities like perception, attention, and memory. However, we think it has been under explored with regard to the social-cognitive abilities that underlie important skills such as cooperation. In this case, we aim understand the social-cognitive abilities and biases that underlie human cooperation, and build them into our AI systems.

For this multi-agent-first AI research program to progress, it needs to settle two main methodological questions: (1) how to train agents so that they develop social-cognitive abilities on par with those of humans; and (2) how to measure progress toward the goal. The *Melting Pot* methodology we describe in this report addresses both of these questions.

Melting Pot: the evaluation protocol

Let’s start first by talking about how Melting Pot provides a recipe for measuring progress. The key idea is to always measure generalization, just as it is in other parts of machine learning ([Cobbe et al., 2019](#); [Farebrother et al., 2018](#); [Hastie et al., 2009](#); [Zhang et al., 2018](#)). If we don’t measure generalization then we run the risk of fooling ourselves into thinking we have made progress despite agents merely becoming increasingly overfit to one another’s behavior ([Carroll et al., 2019](#); [Vezhnevets et al., 2020](#); [Vinyals et al., 2019](#)). Such overfitting can be extremely brittle and unlikely to work in the real world. To measure social-cognitive skills and to simulate questions of human compatibility, the key type of generalization we need to look at is social generalization, i.e. generalization to interactions in unfamiliar social situations involving both familiar and unfamiliar individuals ([Leibo et al., 2021](#)).

Melting Pot consists of a set of *test scenarios* and a protocol for using them. A scenario is a multi-agent environment that evaluates the ability of a *focal population* of agents to generalize to novel social situations. Each scenario is formed by a *substrate* and a *background population*. The substrate is the physical part of the world: its spatial layout, where the objects are, how they move, the rules of physics, etc. The background

population is the part of the simulation that is imbued with agency—excluding the focal population. While the focal population experiences the substrate during training, it has never experienced any of the individuals in the background population. Thus the performance of the focal population in a test scenario measures how well its agents generalize to social situations they were not directly exposed to at training time.

Since Melting Pot seeks to isolate social-environment generalization, it assumes that agents are familiar with the substrate. They have unlimited access to it during training. The evaluation scheme concentrates on assessing the ability of agents (and populations composed thereof) to cope with the presence of unfamiliar individuals in the background population (in a zero-shot way), typically mixed in with familiar individuals from the population under test. If the mixture contains more focal individuals than background individuals we call it a *resident* mode scenario. Whereas, if the mixture contains more background individuals than focal individuals, we call it a *visitor* mode scenario. The name Melting Pot is inspired by viewing this mixing of two distinct populations as analogous to that which occurs in real life when many members of different cultures immigrate to the same destination.

We use the term *multi-agent population learning algorithm* (MAPLA) to refer to any training process that produces a decentralized population of agents capable of simultaneous interaction with one another. The class of MAPLA algorithms is very broad. Most multi-agent reinforcement learning approaches can be made to produce populations. For instance self-play schemes like those used for AlphaGo (Silver et al., 2016, 2017), AlphaZero (Silver et al., 2018), FTW (Capture the Flag) (Jaderberg et al., 2019), hide and seek (Baker et al., 2019), and AlphaStar (Vinyals et al., 2019) fit in the class of MAPLAs, as does recent work on DOTA (Berner et al., 2019) and MOBA (Ye et al., 2020) games, as well as algorithms like MADDPG (Lowe et al., 2017), LOLA (Foerster et al., 2018), PSRO (Lanc-tot et al., 2017), PSRO_N (Balduzzi et al., 2019), and Malthusian reinforcement learning (Leibo et al., 2019). It includes both “centralized train-

ing and decentralized execution” algorithms and “fully decentralized” (decentralized at both training and test time) algorithms. Melting Pot evaluates MAPLAs by evaluating the generalization of populations they produce.

How to train MAPLAs that will perform well in Melting Pot

Now lets talk about the training side of Melting Pot. Researchers and the agents they train are allowed unlimited access to the substrate. This means that at test time the agents will usually already be familiar with their physical environment. So the kind of generalization that Melting Pot probes is mainly along social dimensions, not physical dimensions. This is in contrast to most work on generalization in reinforcement learning (e.g. Cobbe et al. (2019)), which are primarily concerned with generalization along physical dimensions.

In thinking about training for Melting Pot, our guiding principle has been that we shouldn’t make decisions that restrict the scope for creativity in how researchers can train their agents. We can get away with few restrictions on training because we have a rigorous evaluation. As long as researchers do not cheat by training their agents on the test set, then we should be able to say of training that “anything goes”. All agents will be judged according to a common benchmark in the end.

One interpretation of Melting Pot-compatible training processes, which we especially like, is that they give groups of agents the opportunity to form their own artificial civil society, complete with their own conventions, norms, and institutions. Forming a civil society entails the genesis and successful resolution of all kinds of obstacles to cooperation. The specific obstacles that arise, and the ways they can be resolved, depend on properties of the substrate. For instance, free riding in public-good provision is one obstacle to cooperation, and unsustainable resource usage leading to environmental degradation is quite another (Ostrom, 2009). The challenge of overcoming these obstacles is what pushes agents to develop their social-cognitive abilities. For instance, if a particular social dilemma can be resolved by

inventing a concept of fairness and enforcing its associated norms then that very fact motivates agents capable of representing the fairness concept and its norms to actually do so.

Training and testing

A *substrate* is an N -player partially observable general-sum Markov game (Leibo et al., 2021). Previously, in Melting Pot 1.0, we only fully supported symmetric games so each player received observations from the same observation space \mathcal{X} , and took actions from the same action space \mathcal{A} . Melting Pot 2.0 now adds full support for asymmetric games. We implement it by adding one new concept, that a player may have a *role*.

Roles

Here, we extend the definition of a substrate to explicitly allow for situations where players do not all have the same role (e.g. for asymmetric games). Now a player’s observation and action space may vary according to their *role*. Substrates have a set of allowed roles \mathcal{R} . The player in slot i is assigned role $r_i \in \mathcal{R}$, which determines their observation space ($\mathcal{X}_i \equiv \mathcal{X}_{r_i}$) and action space ($\mathcal{A}_i \equiv \mathcal{A}_{r_i}$).

Substrate factories

We define a *substrate factory* $F(\mathbf{r})$, a parameterized function that returns a substrate conditional on a particular *role configuration*, $\mathbf{r} = (r_1, \dots, r_N) \in \mathcal{R}_F^N$, where the number of players supported by the returned substrate is $|\mathbf{r}|$. In principle, F can support any number of players, but in practice it is constrained to $N_F^{\min} \leq |\mathbf{r}| \leq N_F^{\max}$.

The substrate factory may be used freely to sample a substrate, allowing agents to be trained for each specific role supported by that substrate factory. This results in a *focal population*, $\pi \sim f_F(\cdot | \mathbf{r})$; a distribution of policies conditioned on role $r \in \mathcal{R}_F$. It is this conditional distribution that is submitted for evaluation on test scenarios.

Scenarios

A *test scenario* is a Markov game parameterized by a substrate factory F , a role configuration \mathbf{r} , the focal-population size $N \leq |\mathbf{r}|$, and a *background population* of held-out bots g_F . Of these only F is known at training time.

To form a scenario we construct a substrate using $F(\mathbf{r})$, and fill all but the first N player slots with policies sampled from the background population, such that $\pi_i \sim g_F(\cdot | r_i)$ for $i > N$. This results in an N -player Markov game.

We use such test scenarios to evaluate any trained focal population f_F , by filling the scenario’s remaining N slots with policies sampled from the focal population, such that $\pi_i \sim f_F(\cdot | r_i)$ for $i \leq N$. We assess the performance of the focal population on the test scenario by the *focal per-capita return*, the mean return obtained across the focal players (i.e. excluding the background players).

Since the role configuration and background population are not known at training time, the focal per-capita return assesses the ability of the focal population to generalize to novel social situations.

When focal players outnumber background players we say the test scenario is in *resident* mode. These scenarios usually test the emergent cooperative structure of the population under evaluation for its robustness to interactions with a minority of unfamiliar individuals not encountered during training. When background players outnumber focal players, we say the test scenario is in *visitor* mode. One common use case for visitor-mode scenarios in Melting Pot is to test whether an individual from the focal population can observe the conventions and norms of the dominant background population and act accordingly (without retraining).

Secondary evaluation metrics

We propose to use *focal-population per-capita return* as our primary evaluation metric for testing the performance of a learning algorithm in a novel social situation. This is because, first and foremost, we want Melting Pot to provide a rigorous



Figure 1 | The Melting Pot suite contains a set of over 50 multi-agent reinforcement learning substrates on which to train agents, and over 250 unique test scenarios to evaluate them. Screenshots in this figure were taken from the third person visualization view. All substrates are partially observed so agents view them through a smaller egocentric viewing window.

and clearly interpretable evaluation metric that highlights unsolved problems and compares innovative algorithms to one another (Leibo et al., 2021). Melting Pot is wide enough that many other reasonable-sounding metrics do not work well across the entire suite. For instance, we cannot always use the total collective return (sum of all players’ rewards including both focal and background players) because its value is always zero in zero-sum substrates.

However, when evaluating the suitability of trained agents for a practical application, there will be additional considerations, which can be assessed from secondary evaluation metrics using the same test scenarios. For example, impacts on the background population may be an indicator of the impacts the trained agents might have on humans in the real world. So we can measure the *background-population per-capita return* to see if it is negatively impacted by the introduction of the focal population. This could be useful to study whether the joint policy of the focal agents produces negative externalities—“side effects” that impact the broader population while sparing the focal population, dovetailing well with research on value alignment and AI safety (Amodei et al., 2016; Soares and Fallenstein, 2014). Or following Perolat et al. (2017), we can measure the *inequality* of the background-population individual

returns to see if any benefit or harm arising from having introduced the focal population is fairly shared, perhaps connecting fruitfully to beneficial and cooperative AI research agendas (Dafoe et al., 2020; Russell et al., 2015).

Benchmark experiments

Agent architectures

To demonstrate the use of Melting Pot, we provide benchmark MARL results for a number of MAPLA (agent) architectures.

For each MAPLA architecture we performed 3 training runs on each substrate. For most substrates we trained N individual agents in each training run: one in each role of the substrate’s N -player *default role configuration*. The exceptions to this were some 2-player substrates where we trained more than 2 agents:

1. Bach or Stravinsky in the Matrix Repeated (2 agents in the “bach_fan” role, and 2 agents in the “stravinsky_fan” role);
2. other “in the Matrix Repeated” substrates (5 agents in the “default” role);
3. Running with Scissors in the Matrix One Shot (5 agents in the “default” role);
4. Coins (5 agents in the “default” role);

Figure 2 | Multi-agent concepts engaged by each substrate. Properties highlighted here are only intended as a rough guide to the concepts at play in each substrate. They should not be taken as a serious theoretical attempt to classify multi-agent situations.

For each training episode, we built the substrate by passing a random permutation of the *default role configuration* to the substrate factory. To fill each player slots, we sampled *without replacement* from the subpopulation of agents that supported the corresponding role. This ensured that each training episode always had the same number of players with the same roles. Furthermore, each agent played as at most one player in each episode, always played in the same role, and that each agent's player index varied between episodes. Each agent was trained for 10^9 steps.

At test time, we used all trained agents as the focal population in the test scenarios. To fill each focal player slot, we sampled *with replacement* from the subpopulation of agents that supported the corresponding focal-player role. Thus at test time, an agent could play as multiple players in the same test episode.

The RL architectures we trained were: Actor-Critic Baseline (ACB), V-MPO ([Song et al., 2020](#)), and OPRE ([Vezhnevets et al., 2020](#)). OPRE was

specifically designed for MARL while ACB and V-MPO are generic RL algorithms.

ACB is a well-established off-the-shelf RL method. Its core reinforcement learning algorithm was proposed by Espeholt et al. (2018), which built on prior work from Mnih et al. (2016) that described an algorithm called “advantage actor critic” (A3C). We also augmented it with a contrastive predictive coding auxiliary objective to promote discriminability of nearby timepoints using its LSTM state representations (Oord et al., 2018). In the previous Melting Pot release this agent was called A3C (Leibo et al., 2021). It has been the baseline RL agent used in a few papers in this line of research, and has nearly always been called either A3C or A2C e.g. Bakker et al. (2021); Duéñez-Guzmán et al. (2021); Köster et al. (2020, 2022); McKee et al. (2020); Vinitsky et al. (2023). We chose to rename it now because the old name has proven confusing in some contexts, and to reflect that it has departed quite significantly from the one called A3C in Mnih et al.

(2016). We chose the new name, ACB, to evoke the old name while still clearly being something different. It still starts with A, and remains three letters long. It even rhymes with A3C. It stands for Actor-Critic Baseline. We plan to use continue using this name going forward.

Since the last Melting Pot release we added a PopArt module to both ACB and OPRE (Hessel et al., 2019). V-MPO was already using PopArt. This layer is attached to the baseline (value) and advantage computation and serves to normalize the gradients. It helps avoid problems that would otherwise arise when reward values vary through several orders of magnitude over the course of training and allows us to use the same hyperparameters for all substrates, even though they have very different reward scales from one another.

We also trained *prosocial* variants of both algorithms, which directly optimized the per-capita return (rather than individual return), by sharing reward between players during training. Optimizing for collective return as a surrogate objective has previously been used for collaborative games (substrates) and social dilemmas (Claus and Boutilier, 1998; Peysakhovich and Lerer, 2017), and our experiments here allow us to investigate whether it generalizes well.

Computational setup

We use a distributed training setup described in Hessel et al. (2021), where TPU handles the agent parameters update in a learner process and action inference (learning and inference are split over several different chips). Experience for learning is generated in an actor process. The actor handles interactions with the environment on a CPU. The episode trajectories are split into truncated sequences each 100 steps long and are grouped into batches, which are then split over 6 TPU chips for learning. Each chip computes the update using the corresponding batch; afterwards their updates are averaged and applied to the parameters. This is different to the setup in Leibo et al. (2021), where learning was done on GPUs and inference was performed on CPU actors directly. This change led to hyper-parameters of both agents having to be re-calibrated.

Agents hyper-parameters

All agent architectures had the same size convolutional net and LSTM. The OPRE agent had additional hierarchical structure in its policy as described in Vezhnevets et al. (2020). All agents had a pop-art layer (Hessel et al., 2019) for normalizing the value function. ACB minimized a contrastive predictive coding loss (Oord et al., 2018) as an auxiliary objective (Jaderberg et al., 2016) to promote discrimination between nearby time points via LSTM state representations (a standard augmentation in recent work with ACB). The size of layers and their structure was the same as in the original Melting Pot release (Leibo et al., 2021). ACB and V-MPO only differ by the loss. For V-MPO loss we used default hyper-parameters from Song et al. (2020).

We tuned hyper-parameters of each agent on a limited set of substrates by optimizing for the collective return in population play and focal return when trained as an exploiter—we did not use the generalisation score for tuning. We have done it to the best of our ability given a limited computational budget.

All agents used discount rate $\gamma = 0.99$, PopArt step size of 10^{-3} with lower bound on scale 10^{-2} upper bound of 10^6 and gradient normalisation to 2.0 (Pascanu et al., 2013).

ACB The agent’s network consists of a convNet with two layers with 16, 32 output channels, kernel shapes 8, 4, and strides 8, 1 respectively. It is followed by an MLP with two layers with 64 neurons each. All activation functions are ReLU. It is followed by an LSTM with 128 units. Policy and baseline (for the critic) are produced by linear layers connected to the output of LSTM, followed by PopArt. We used an auxiliary loss (Jaderberg et al., 2016) for shaping the representation using contrastive predictive coding (Oord et al., 2018). CPC was discriminating between nearby time points via LSTM state representations (a standard augmentation in recent works with A3C). We used RMSProp optimizer with learning rate of $4 * 10^{-4}$, $\epsilon = 10^{-5}$, momentum set to zero, decay of 0.99, and the batch size of 256. Baseline cost was 1.0, entropy regularizer for policy at 0.003.

V-MPO We used RMSProp optimizer with learning rate of $8 * 10^{-4}$, $\epsilon = 10^{-5}$, momentum set to zero, decay of 0.99, and the batch size of 256. Initial $\eta = 1$, $\alpha = 5$, $\epsilon_{KL} = 0.01$. The gradients were clipped at 2.0 and target network was updated every 10 update steps.

OPRE The networks shares the basic structure with ACB. The details of the architecture can be found in Vezhnevets et al. (2020). We used 16 options and a hierarchical critic implemented as an MLP with 64, 64 neurons. Critic has access to opponents observations. In Vezhnevets et al. (2020) only one other player was present. To represent multiple co-players, we simply average outputs of the visual stack (convNet+MLP) and their inventories (where appropriate) together, before feeding it into the critic. OPRE had the same parameters for optimisation as ACB plus two extra regularizers: i) entropy of policy over options set to 0.04 and ii) KL cost between critic and actor policies over options set at 0.01. Learning rate was set to $8 * 10^{-4}$ and the batch size of 96

Puppet for held-out bots: Sometimes the desired behavior for held-out bots is difficult to specify using a single reward function. In these cases, we generate background populations using techniques inspired by hierarchical reinforcement learning (Sutton et al., 1999, 2011); in particular reward shaping (Sutton and Barto, 2018) and “option keyboard” (Barreto et al., 2019). We create a basic portfolio of behaviors by training bots that use different environment events as the reward signal (as in the Horde architecture from Sutton et al. (2011)). We then chain them together using simple Python code. This allows us to express complex behaviours in an “if this event, run that behaviour” way. For example, in *Clean Up* we created a bot that only cleans if other players are cleaning. These bots had a special network architecture based on FuN (Vezhnevets et al., 2017), with goals specified externally via substrate events rather than being produced inside the agent. The idea is to use environment events as goals for training basic policies and then combine them into complex ones using a pre-scripted

high-level policy. The high-level policy is a set of “if this event happens, activate that behaviour” statements. The same set of environment events can be used to train basic policies and script the high-level one. We call this approach *Puppet*.

The puppet neural network has the same basic structure as the other agents we tested, input→ConvNet→MLP→LSTM→outputs, but has a hierarchical policy structure. The architecture was inspired by Feudal Networks (Vezhnevets et al., 2017), but has several important differences. We represent goals as a one-hot vector g , which we embed into a continuous representation $e(g)$. We then feed e as an extra input to the LSTM. The network outputs several policies $\pi_z(a|x)$ and the final policy is a mixture $\pi(a|x) = \sum_z \alpha(e)\pi_z(a|x)$, where the mixture coefficients $\alpha(e) = \text{SoftMax}(e)$ are learnt from the embedding. Notice, that instead of directly associating policies to goals, we allow the embedding to learn it through experience. To train the puppet to follow goals, we train it in the respective environment with goals switching at random intervals and rewarding the agent for following them. Each goal has a separate value function and a PopArt module.

Normalising results

We also use two special MAPLAs for normalization: “exploiters” and “random”. They are intended to provide rough upper and lower bounds for the raw scores. Though, as we will discuss below, they do not always operate entirely as intended, and this has implications for how the normalized results should be interpreted (Fig. 3).

Each exploiter is a self-interested ACB or OPRE MAPLA trained directly on a single test scenario, using their individual return as the reward signal without further augmentation. Exploiters trained for 10^9 steps (same as the MAPLAs under evaluation). The random agent selects actions uniformly at random, and ignores input observations. Together, exploiters and the random agent provide a rough estimate of the upper and lower bounds (respectively) of performance on the test scenarios. To contextualize the range of agent returns, we min-max normalize the focal per-capita re-

| | exploiter | acb | vmpo | opre | acb_prosocial | opre_prosocial | random | overall score |
|-------------------------------------------------------------|-----------|------|------|------|---------------|----------------|--------|-------------------------------------------------------------|
| overall score | 0.88 | 0.55 | 0.52 | 0.45 | 0.35 | 0.28 | 0.03 | overall score |
| allelomorphic_harvest_open | 0.96 | 0.98 | 0.83 | 0.77 | 0.38 | 0.42 | 0.00 | allelomorphic_harvest_open |
| bach_or_stravinsky_in_the_matrix_arena | 1.00 | 0.83 | 0.78 | 0.79 | 0.66 | 0.55 | 0.00 | bach_or_stravinsky_in_the_matrix_arena |
| bach_or_stravinsky_in_the_matrix_repeated | 1.00 | 0.65 | 0.63 | 0.55 | 0.68 | 0.63 | 0.00 | bach_or_stravinsky_in_the_matrix_repeated |
| boat_race_eight_races | 1.00 | 0.37 | 0.68 | 0.00 | 0.01 | 0.00 | 0.00 | boat_race_eight_races |
| chemistry_three_metabolic_cycles | 1.00 | 0.17 | 0.16 | 0.08 | 0.03 | 0.01 | 0.00 | chemistry_three_metabolic_cycles |
| chemistry_three_metabolic_cycles_with_plentiful_distractors | 1.00 | 0.34 | 0.34 | 0.17 | 0.10 | 0.04 | 0.00 | chemistry_three_metabolic_cycles_with_plentiful_distractors |
| chemistry_two_metabolic_cycles | 1.00 | 0.66 | 0.17 | 0.22 | 0.12 | 0.00 | 0.00 | chemistry_two_metabolic_cycles |
| chemistry_two_metabolic_cycles_with_distractors | 0.77 | 0.74 | 0.17 | 0.09 | 0.13 | 0.03 | 0.00 | chemistry_two_metabolic_cycles_with_distractors |
| chicken_in_the_matrix_arena | 0.96 | 0.92 | 0.87 | 0.91 | 0.95 | 0.88 | 0.00 | chicken_in_the_matrix_arena |
| chicken_in_the_matrix_repeated | 0.97 | 0.73 | 0.70 | 0.72 | 0.81 | 0.80 | 0.00 | chicken_in_the_matrix_repeated |
| clean_up | 0.71 | 0.05 | 0.03 | 0.02 | 0.63 | 0.02 | 0.02 | clean_up |
| coins | 0.93 | 0.84 | 0.81 | 0.82 | 0.82 | 0.80 | 0.00 | coins |
| collaborative_cooking_asymmetric | 0.67 | 0.39 | 0.65 | 0.14 | 0.39 | 0.14 | 0.00 | collaborative_cooking_asymmetric |
| collaborative_cooking_circuit | 1.00 | 0.35 | 0.34 | 0.34 | 0.35 | 0.34 | 0.35 | collaborative_cooking_circuit |
| collaborative_cooking_cramped | 0.67 | 0.18 | 0.30 | 0.35 | 0.18 | 0.35 | 0.06 | collaborative_cooking_cramped |
| collaborative_cooking_crowded | 0.86 | 0.46 | 0.52 | 0.49 | 0.46 | 0.49 | 0.56 | collaborative_cooking_crowded |
| collaborative_cooking_figure_eight | 1.00 | 0.35 | 0.35 | 0.34 | 0.35 | 0.34 | 0.34 | collaborative_cooking_figure_eight |
| collaborative_cooking_forced | 1.00 | 0.02 | 0.28 | 0.02 | 0.02 | 0.02 | 0.02 | collaborative_cooking_forced |
| collaborative_cooking_ring | 1.00 | 0.00 | 0.82 | 0.00 | 0.00 | 0.00 | 0.00 | collaborative_cooking_ring |
| commons_harvest_closed | 0.60 | 0.53 | 0.81 | 0.80 | 0.64 | 0.32 | 0.00 | commons_harvest_closed |
| commons_harvest_open | 0.84 | 0.83 | 0.89 | 0.81 | 0.94 | 0.52 | 0.00 | commons_harvest_open |
| commons_harvest_partnership | 0.82 | 0.66 | 0.68 | 0.63 | 0.77 | 0.44 | 0.00 | commons_harvest_partnership |
| coop_mining | 1.00 | 0.52 | 0.44 | 0.41 | 0.27 | 0.12 | 0.00 | coop_mining |
| daycare | 0.97 | 0.87 | 0.83 | 0.60 | 0.36 | 0.30 | 0.00 | daycare |
| externality_mushrooms_dense | 0.79 | 0.56 | 0.53 | 0.53 | 0.68 | 0.61 | 0.00 | externality_mushrooms_dense |
| factory_commons_either_or | 0.97 | 0.57 | 0.34 | 0.02 | 0.06 | 0.00 | 0.00 | factory_commons_either_or |
| fruit_market_concentric_rivers | 0.98 | 0.97 | 0.32 | 0.78 | 0.12 | 0.02 | 0.00 | fruit_market_concentric_rivers |
| gift_refinements | 0.67 | 0.16 | 0.16 | 0.16 | 0.45 | 0.10 | 0.05 | gift_refinements |
| hidden_agenda | 0.67 | 0.40 | 0.34 | 0.30 | | | 0.05 | hidden_agenda |
| paintball_capture_the_flag | 1.00 | 0.03 | 0.03 | 0.03 | | | 0.00 | paintball_capture_the_flag |
| paintball_king_of_the_hill | 0.25 | 0.91 | 0.57 | 0.61 | | | 0.00 | paintball_king_of_the_hill |
| predator_prey_alley_hunt | 0.88 | 0.92 | 0.80 | 0.37 | 0.12 | 0.04 | 0.00 | predator_prey_alley_hunt |
| predator_prey_open | 0.86 | 0.80 | 0.66 | 0.93 | 0.08 | 0.03 | 0.00 | predator_prey_open |
| predator_prey_orchard | 0.94 | 0.79 | 0.74 | 0.69 | 0.06 | 0.03 | 0.00 | predator_prey_orchard |
| predator_prey_random_forest | 0.50 | 0.96 | 0.77 | 0.54 | 0.02 | 0.02 | 0.00 | predator_prey_random_forest |
| prisoners_dilemma_in_the_matrix_arena | 0.87 | 0.85 | 0.81 | 0.83 | 0.94 | 0.87 | 0.00 | prisoners_dilemma_in_the_matrix_arena |
| prisoners_dilemma_in_the_matrix_repeated | 0.93 | 0.69 | 0.66 | 0.67 | 0.60 | 0.52 | 0.00 | prisoners_dilemma_in_the_matrix_repeated |
| pure_coordination_in_the_matrix_arena | 0.99 | 0.77 | 0.72 | 0.75 | 0.55 | 0.53 | 0.00 | pure_coordination_in_the_matrix_arena |
| pure_coordination_in_the_matrix_repeated | 1.00 | 0.53 | 0.44 | 0.50 | 0.47 | 0.49 | 0.00 | pure_coordination_in_the_matrix_repeated |
| rationalizable_coordination_in_the_matrix_arena | 0.95 | 0.85 | 0.78 | 0.74 | 0.66 | 0.57 | 0.00 | rationalizable_coordination_in_the_matrix_arena |
| rationalizable_coordination_in_the_matrix_repeated | 1.00 | 0.51 | 0.39 | 0.54 | 0.48 | 0.51 | 0.00 | rationalizable_coordination_in_the_matrix_repeated |
| running_with_scissors_in_the_matrix_arena | 1.00 | 0.01 | 0.06 | 0.02 | | | 0.01 | running_with_scissors_in_the_matrix_arena |
| running_with_scissors_in_the_matrix_one_shot | 0.84 | 0.12 | 0.17 | 0.21 | | | 0.02 | running_with_scissors_in_the_matrix_one_shot |
| running_with_scissors_in_the_matrix_repeated | 1.00 | 0.01 | 0.01 | 0.01 | | | 0.01 | running_with_scissors_in_the_matrix_repeated |
| stag_hunt_in_the_matrix_arena | 0.87 | 0.72 | 0.87 | 0.70 | 0.80 | 0.67 | 0.00 | stag_hunt_in_the_matrix_arena |
| stag_hunt_in_the_matrix_repeated | 0.90 | 0.77 | 0.76 | 0.76 | 0.69 | 0.67 | 0.00 | stag_hunt_in_the_matrix_repeated |
| territory_inside_out | 0.76 | 0.48 | 0.69 | 0.40 | 0.09 | 0.25 | 0.03 | territory_inside_out |
| territory_open | 0.94 | 0.39 | 0.53 | 0.39 | 0.10 | 0.07 | 0.00 | territory_open |
| territory_rooms | 0.99 | 0.69 | 0.57 | 0.68 | 0.24 | 0.32 | 0.06 | territory_rooms |

Figure 3 | MAPLA performance scores averaged over each substrate's scenarios. Averaged over 3 independent runs for each MAPLA.

turns to get a *performance score* that is between 0 (for the worst agent) and 1 (for the best agent). Often the score of 0 corresponds to the random agent and 1 to the exploiter. However, for some scenarios this is not the case, since held-out bots might create a task too hard to solve for a naive agent. This clearly manifests in competitive substrates, but is not limited to them. Note that this makes the normalized scores of the MAPLAs reported in Fig. 3 appear artificially high in those cases.

Below is the list of scenarios where the exploiter performed as no better than a random agent:

1. *Clean Up*: scenario 1;
2. *Collaborative Cooking Asymmetric / Circuit / Cramped / Figure Eight*: scenario 2;
3. *Commons Harvest Closed*: scenario 3;
4. *Paintball King of the Hill*: scenarios 0, 1, 3;
5. *Predator Prey Random Forest*: scenario 3;
6. *Running with Scissors in the Matrix Repeated*: scenarios 0, 1, 2, 3, 4 for ACB exploiter was no better than random. We trained an OPRE exploiter as replacement, which achieves strong results.
7. *Territory Inside Out*: scenario 1;

On *Collaborative Cooking* scenario 0 the random agent performed better than any other agent. This is due to the random agent “staying out of the way” of the competent held-out bots, whereas agents produced by MAPLAs interfered with them. In *Territory Rooms* scenario 3 and *Territory Inside-out* scenario 5 the random agent performs better than some, but worse than most agents.

On *Coins* (all scenarios), the normalization procedure yields spuriously strong results for a different reason. Here the random agent gets a dramatically worse score than any other agent. Its score is a large negative number; and this is an environment which would otherwise usually produce values around zero for defection equilibria or large positive numbers in the case of mutual cooperation. The random agent performs so poorly that the normalization makes it look like all the MAPLAs we evaluated perform well by comparison. However, visually inspecting their policies shows that that is not the case. None of

the MAPLAs we tested actually learned to cooperate in the *Coins* substrate.

Results summary

Fig. 3 presents an overview, showing the average test scenario score for each substrate and each MAPLA. Please note that the presented results should not be interpreted as a definitive statement on the performance of specific MAPLAs. We didn’t perform careful tuning for each MAPLA, neither it was our goal to provide the state-of-the-art results. These results should be interpreted as the reasonable performance ballpark of off-the-shelf RL methods. We encourage researchers to run their own baselines.

On average, the top performing agent was ACB, followed V-MPO, and then by OPRE. This is different to our initial findings in the first Melting Pot paper (Leibo et al., 2021), where the ranking was V-MPO, OPRE, ACB. Our main hypothesis for ACB pulling ahead is that it greatly benefited from PopArt module. Most likely, V-MPO was doing better in the original experiments (Leibo et al., 2021) thanks to PopArt (which ACB did not use in the original paper, but V-MPO did). Now, the rather more sophisticated V-MPO algorithm seems to perform worse than the much simpler ACB policy gradient algorithm with CPC.

There are several potential reasons for under-performance of OPRE it: i) PopArt modules might have benefited ACB more than OPRE ii) scenarios added since the first release are harder for OPRE iii) changes in computational setup affected the two MAPLAs differently. Moreover, we used a very simple way to integrate information over multiple opponent (average pooling) for OPRE, which is likely to limit its capabilities for games with large number of players.

Overall, prosocial agents underperformed their purely self-interested counterparts, but the picture is nuanced. Optimizing for per-capita return can be difficult because it complicates credit assignment, and creates spurious reward “lazy agent” problems (Rashid et al., 2018; Sunehag et al., 2018). However, there are situations where prosocial agents outperform. For instance, in the relatively harsh social dilemma *Clean Up*, only

prosocial agent architectures managed to learn policies that were significantly better than random. Prosocial agents perform best on several other substrates too. These include *Prisoners Dilemma / Stag Hunt in The Matrix*, and *Commons Harvest Open / Partnerships*. This suggests a MAPLA capable of doing well on all of Melting Pot would have to generate agents who can flexibly balance self interest and prosociality depending on the circumstances.

On several groups of substrates the MAPLAs we ran failed to generalize at all, getting scores very close to random: *Collaborative Cooking*, *Running with Scissors*, *Paintball Capture the Flag*. In most other cases generalisation scores were above random, but with a significant room for further improvement.

Train & test scores as a function of training time

See learning curve figures (Figs. 6–54) for each focal population’s per capita return as a function of training time. In each figure the plot on the far left shows results from the training environment (i.e. only focal players). We periodically saved checkpoints of the focal population after every x training steps. All plots to the right of the training plot show result from testing the saved focal population checkpoints from each x in a specific test scenario. The curves plot the mean episode reward and its standard deviation, averaged over 3 independent runs.

Discussion

Here we have presented Melting Pot 2.0: an evaluation suite for MAPLAs that evaluates generalization to novel social situations. This release more than doubles the size of the initial Melting Pot suite (Leibo et al., 2021). There are more than twice as many substrates as there were in the original set and more than three times as many scenarios. To the best of our knowledge, it is the largest open-source benchmark for general-sum Markov games. Moreover, Melting Pot is far from a solved problem. Our preliminary results show that there is plenty of room to improve the

scores in the vast majority of test scenarios and substrates.

To facilitate secondary analyses of published results, we have included standardized evaluation scripts along with Melting Pot 2.0 (see <https://github.com/deepmind/meltingpot>). Other fields such as medicine extensively employ secondary analysis techniques like meta-analysis (Haidich, 2010). However, machine learning has been slow to adopt these methods. They require a bit of foresight on the part of authors to report their results in standard formats, but the whole community benefits when secondary analyses are easier to perform (Burnell et al., 2023; Roelofs et al., 2019). For instance, meta-analysis may be used to quantify convergence between different approaches and clarify rates of improvement over time (Hernández-Orallo, 2017; Martínez-Plumed et al., 2020). Given instance-level data, they can be used to identify blind spots where further investment may be needed (Martínez-Plumed et al., 2019). Encouraging the standardized reporting of experimental results is complementary to other open science efforts like those that encourage public code. Hopefully it will be easy for researchers using Melting Pot to report results in such a way as to encourage such downstream re-use of their results.

Melting Pot 2.0 allows researchers to evaluate algorithms that train populations of agents to behave socially. How will the resulting agents resolve social dilemmas? Can they deal with free-riders? Will they learn policies that have an equitable distribution of returns? Will the winner take all? These questions not only present interesting research directions, but are also important to consider in the context of AI safety and fairness. After all, human society is a multi-agent system and any deployed AI agent will have to generalize to it.

Substrates and Scenarios

Common properties shared by all substrates

Unless otherwise stated, all substrates have the following common rules:

- Sprites are 8×8 pixels.
- The agents have a partial observability window of 11×11 sprites, offset so they see more in front than behind them. The agent sees 9 rows in front of itself, 1 row behind, and 5 columns to either side. A few substrates use smaller observation windows, noted in their respective sections.
- Thus in RGB pixels, the size of each observation is $88 \times 88 \times 3$. All agent architectures used here have RGB pixel representations as their input.
- Movement actions are: forward, backward, strafe left, strafe right, turn left, and turn right.

Allelopathic Harvest¹

There are three varieties of berry: red, green, and blue, and a fixed number of berry patches, which can be replanted to grow any variety of berry. The ripening rate of each berry variety depends linearly on the fraction that that variety (color) comprises of the total. This can be understood as the cumulative effect of each berry variety suppressing the two others. Thus a monoculture of one variety (100% of patches planted with that variety) implies a zero ripening rate for the other two varieties and the highest possible ripening rate for the monoculture variety. Evenly proportioned varieties ($1/3, 1/3, 1/3$) yield the lowest possible overall ripening rate since all varieties suppress one another equally. Thus the sixteen players can increase the overall ripening rate by planting in such a way as to move the total proportions closer to monoculture of either red, green, or blue. However, increasing one variety's proportion decreases the other two since the three proportions must add up to one.

Players in Allelopathic Harvest have heterogeneous tastes: each has a particular berry variety that is intrinsically more rewarding to them. This creates tensions between groups of players since it is only possible to build one monoculture at a time. In addition, there is also a free-rider problem since individuals prefer to consume than plant (Köster et al., 2020). If all individuals free

ride (none ever plant), then the variety proportions remain at their evenly mixed value where they start each episode, so the ripening rate remains very slow.

Individuals are rewarded for consuming ripe berries. They get a reward of 2 for consuming their preferred berry, and a reward of 1 for consuming a berry of any other color. At the start of each episode, berries are initialized to be unripe and evenly distributed over the three colors. Individuals can replant unripe berries to any color. Thus each individual experiences a tension between their incentive to immediately consume ripe berries and their incentive to plant unripe—and their preferred colored—berries. The default player roles are 8 players preferring red berries, and 8 preferring green berries.

The environment (a 29×30 plane) is filled with 348 berry plants of 3 colors (reset at the start of each episode; 116 per color).

Berry ripening depends stochastically on the number of berries sharing the same color that have been planted. Initially all berries are in an unripe state. Each berry has a probability p to ripen on each step, dependent on the number b of berries of the same color across the whole map; $p = 5 \times 10^{-6}b$ (the ‘allelopathic’ mechanic (inspired by Leibo et al. (2019))). Investing in establishing one color throughout the map, a monoculture, is prudent because it can be done relatively rapidly (if all players join in) and by doing so, all players will be able to harvest berries at a much faster rate for the remainder of the episode.

Players can move around in the environment and interact with berries in several ways. Players can use a planting beam to change the color of unripe berries to one of the other berry-type colors (the ‘harvest’ mechanic). Players can also walk over ripe berries to consume them. Ripe berries cannot be replanted, and unripe berries cannot be consumed. Players’ avatars are recolored after using their planting beam to the same color they turned the berry into. Players’ avatars are also stochastically recolored to a fixed white color when they eat a berry (probability inversely proportional to the highest berry fraction). These rules have the effect that past eating/planting ac-

¹For a video of Allelopathic Harvest, see <https://youtu.be/BbOduMGOYF4>

tions often remains visible for others until a new action is taken.

Each player also has the ability to zap other agents with a beam. It could be used either as a punishment mechanism or as a way to compete over berries. Getting zapped once freezes the zapped player for 25 steps and applies a visible mark to the player indicating that they have been punished. If a second zap is received within 50 steps, the player is removed for 25 steps and receives a penalty of -10 reward. If no zap is received for 50 steps, the mark fades. After each use of the zapping beam it is necessary to wait through a cooldown period of 4 steps before it can be used again.

Episodes last 2000 steps. The action space consists of movement, rotation, use of the 3 planting actions, and use of the zapping beam (10 actions total).

Scenarios

SC 0 *Visiting a population where planting green berries is the prevailing convention.* This is a visitor mode scenario with a background population of ACB bots who were selected out of a wider set on the basis of their relatively increased tendency to plant green berries. All four focal players prefer red berries, but since they are outnumbered by the twelve background players who all plant green, they must acquiesce to a convention where green berries become the most common.

SC 1 *Visiting a population where planting red berries is the prevailing convention.* This is a visitor mode scenario with a background population of ACB bots who were selected out of a wider set on the basis of their relatively increased tendency to plant red berries. The four focal players also prefer red berries so they do well under the prevailing convention.

SC 2 *Focals are resident and visited by bots who plant either red or green.* This is a resident mode scenario with a background population of ACB bots who were selected on

the basis of their relatively increased tendency to plant either red or green berries. Of the fourteen focal players, most prefer red berries but some prefer green berries. If the two background players plant inconsistently with the resident/focal population's established convention then members of the focal population should take sanctioning actions to prevent them from planting too much.

Boat Race: Eight Races²

The environment is inspired by the boat rowing thought experiment posed by Hume (1739/2012). Six players engage in a back and forth series of boat races across a river. Boats, however, cannot be rowed by a single player, and thus, players need to find a partner before each race and coordinate their rowing during the race to cross the river. At the beginning of a race, apples which confer reward when eaten appear on the other bank of the river. When the players are on the boat, they can choose from two different rowing actions at each timestamp: (a) *paddle*, which is efficient, but costly if not coordinated with its partner; and (b) *flail*, an inefficient action which isn't affected by the partner's action. When both players paddle simultaneously, the boat moves one place every 3 timesteps toward the other side of the river. When either player *flails*, the boat has a 10% probability of moving one place, regardless of what the other player does. If one of the players on the boat paddles (i.e. if they have executed the paddle action within the last 3 timesteps) and the other flails, the paddler receives a penalty reward of -0.5 . The episode consists of eight races of fixed duration. Each race consists of a partner choice phase where the access to the boats is barred, but players can line up. This phase lasts 75 steps. After the partner choice phase, the actual race starts by lifting the bars preventing players from accessing the boats. Once a player sits (by touching) on a boat seat, they are prevented from moving until the reach the other side of the river. The actual race lasts for 225 steps. Any player that doesn't reach the other

²For a video of Boat Race: Eight Races, see <https://youtu.be/sEh1hRJVuFw>

side of the river when the race ends is disqualified (removed from the episode). The rowing styles are visibly distinct during the race, with mutual rowing showing steady rhythmic progress of the boat, and flailing looking like stochastic progress, and an oar always up.

Scenarios

SC 0 Visiting cooperators. This is a visitor mode scenario with a background population of ACB bots that are incentivized to only paddle when on the boat. A single focal player must find any partner, and then paddle alongside their boat partner in order to maximize the speed to get to the other side, and thus get the most reward. If the focal player flails, on average they will receive lower reward, while causing large negative rewards to their partner.

SC 1 Visiting defectors. This is a visitor mode scenario with a background population of ACB bots that are incentivized to only flail when on the boat. A single focal player must find any partner, and then flail alongside their boat partner. Alternatively, the focal player might choose to completely free-ride, not rowing at all. This is a slightly suboptimal strategy, because any timestep that their partner doesn't flail, there is no chance to make the boat move. The focal player should never paddle, as no partner would likely paddle in return.

SC 2 Visited by a cooperator. This is a resident mode scenario with a background population of a single ACB bot that is incentivized to only paddle when on the boat. Five focal players can partner among themselves, but at least one must partner with the background bot. In that case, they should paddle, as that maximizes their reward. However, flailing is also a reasonable strategy, if a bit less optimal (see Scenario 0 above).

SC 3 Visited by a defector. This is a resident mode scenario with a background population of a single ACB bot that is incentivized to only flail when on the boat. Five focal players can partner among themselves, but at least

one must partner with the background bot. In that case, they should flail, as that maximizes their reward. However, free-riding (not rowing) is also a reasonable strategy, if a bit less optimal (see Scenario 1 above).

SC 4 Find the cooperator partner. This is a visitor mode scenario with a background population of ACB bots, four of which are incentivized to only flail while one is incentivized to paddle when on the boat. A single focal player must find the cooperator partner, and then paddle alongside their boat partner in order to maximize the speed to get to the other side. The focal player has several possible strategies to find the cooperator: they can use rejection sampling, e.g. choosing a partner at random, and then observing their strategy; if they like them, choose them for all subsequent races, if they don't like them, sample someone else. With eight races and five co-players, it should be in principle possible to sample all co-players at least one. Alternatively, the focal player can choose a partner at random for the first race, but orient themselves in such a way that they can observe all the boats during the actual race, and therefore know who the cooperator is. Then, thereafter, always partner with the cooperator.

Chemistry *

Reactions are defined by a graph (see Fig. 4 and Fig. 5), which together with a map setting initial molecules, defines a substrate, occur stochastically when reactants are brought near one another. Agents can carry a single molecule around the map with them at a time. Agents are rewarded when a specific reaction—such as metabolizing food—occurs with the molecule in their inventory participating as either a reactant or a product. Each reaction has a number of reactants and products, occurs at different rates that can depend on if it is in an agent's inventory or not. As an example, metabolizing food in the metabolic cycles substrate has a much higher rate in the inventory where it generates a reward than outside where it represents the food molecule dissipating. Some tasks also contain distractor molecules,

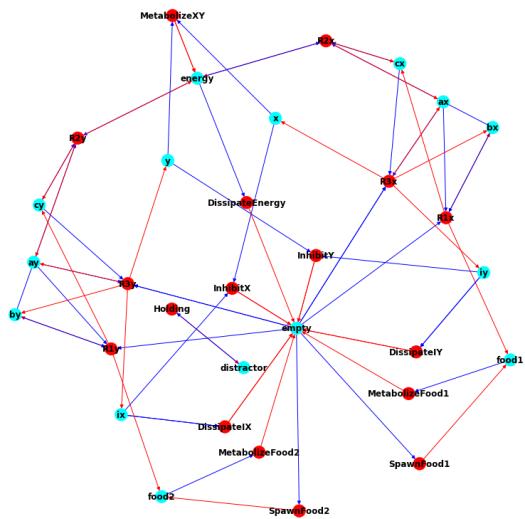


Figure 4 | Reaction Graph for molecules in Chemistry: Two Metabolic Cycles with Distractors. The task variants with or without distractors differ only by the presence of distractor molecules in the map

that allow agents to easily generate a stream of modest rewards by holding such a molecule in their inventory. This strategy is a shallow local optima that is easily found by agents, making it harder to learn to more rewarding cycles.

Chemistry: Three Metabolic Cycles³

Individuals benefit from three different food generating reaction cycles. The cycles will run on their own (autocatalytically), but require energy to continue. One of the cycles produces more rewarding compounds than the other two but it also consumes the energy used to power it at a faster rate. Bringing together side products from the other two cycles generates new energy such that both cycles can continue. If the agents get them running consistently enough then they can generate enough energy from them to also power the third cycle, which is more energy hungry. Too much use of the energy-hungry cycle before discovering how to use the other two cycles efficiently will rapidly deplete the initial energy

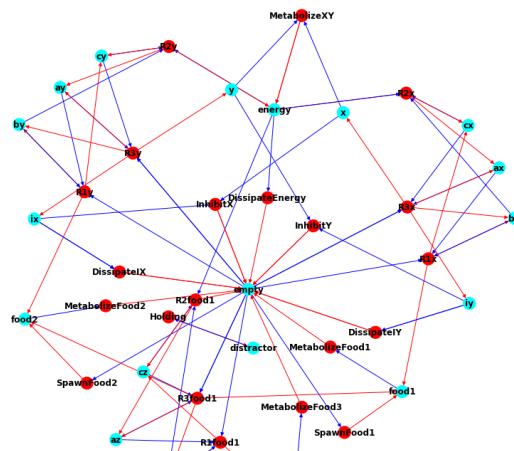


Figure 5 | Reaction Graph for molecules in Chemistry: Three Metabolic Cycles with Plentiful Distractors. The task variants with or without distractors differ only by the presence of distractor molecules in the map

present in the map. The highest collective return is obtained by populations that run all three autocatalytic reaction cycles simultaneously.

Scenarios

SC 0 Resident focal population meets a small mixture of background bots. Five focal agents meet three bots that each specialize in running one of the three cycles. The five focal agents need to contribute to the red and blue metabolic cycles and quickly generate more energy by combining the side products so as to allow for also running the more rewarding yellow cycles.

SC 1 Meeting bots running blue and yellow. Four focal agents meet four bots, two of which runs the blue metabolic and two that runs the yellow. The focal agents need to fully run the green cycle and generate more energy by bringing side products together. They need this part to be efficient enough to not have the two bots running the yellow more energy demanding cycle depleting the energy. They also need to hold on to some

³For a video of *Chemistry: Three Metabolic Cycles*, see <https://youtu.be/RlnojJHAoFI>

energy for running the green and blue cycles and keep that away from the yellow running bots.

- SC 2** *Meeting one-sided bots running green.* Four focal agents meet four bots that all run the green metabolic cycle. The focal agents need to run the blue cycle and generate more energy by bringing side products together. If they do this efficiently enough they can also benefit from mixing in the yellow cycle.
- SC 3** *Visit a resident population with mixed specialties.* Two focal agents meets six bots, two for each of the three cycles. The focal agents needs to generate more energy by bringing side products together and to chip in running the blue and green fast enough for the yellow running bots to not deplete all energy.

Chemistry: Three Metabolic Cycles With Plentiful Distractors⁴

Similar to *Chemistry: Three Metabolic Cycles* but here there are also molecules of a distractor compound scattered around. The distractor can be collected and held, after which it provides reward at a constant but low rate. Holding a distractor is the simplest rewarding strategy, it corresponds to a shallow local optimum that is very easy for agents to learn, and once learned often makes it very difficult to explore other policies. For agents to get good scores in this substrate they must be able to avoid premature convergence to such shallow local optima and continue to explore beyond them.

Scenarios

- SC 0** *Resident focal population meets a small mixture of background bots, must avoid distractor molecules.* Five focal agents meet three bots that each specialize in running one of the three cycles. The five focal agents needs to contribute to the red and blue metabolic cycles and quickly generate more energy by

⁴For a video of *Chemistry: Three Metabolic Cycles With Plentiful Distractors*, see <https://youtu.be/IjlJckwM1VE>

combining the side products so as to allow for also running the more rewarding yellow cycles. The focal agents also needs to avoid the distractor molecules that are not touched by the bots.

- SC 1** *Meeting bots running blue, avoid distractors.* Four focal agents meet four bots that all run the blue metabolic cycle. The focal agents need to run the green cycle and generate more energy by bringing side products together. If they do this efficiently enough they can also benefit from mixing in the yellow cycle. The focal agents also needs to avoid the distractor molecules that are not touched by the bots.
- SC 2** *Meeting bots running green and yellow, avoid distractors.* Four focal agents meet four bots, two of which runs the green metabolic and two that runs the yellow. The focal agents need to fully run the blue cycle and generate more energy by bringing side products together. They need this part to be efficient enough to not have the two bots running the yellow more energy demanding cycle depleting the energy. They also need to hold on to some energy for running the green and blue cycles and keep that away from the yellow running bots. The focal agents also needs to avoid the distractor molecules that are not touched by the bots.
- SC 3** *Visit a resident population with mixed specialties and avoid distractor molecules.* Two focal agents meets six bots, two for each of the three cycles. The focal agents needs to generate more energy by bringing side products together and to chip in running the blue and green fast enough for the yellow running bots to not deplete all energy. The focal agents also needs to avoid the distractor molecules that are not touched by the bots.

Chemistry: Two Metabolic Cycles⁵

Individuals benefit from two different food generating cycles of reactions that both rely on energy

⁵For a video of *Chemistry: Two Metabolic Cycles*, see <https://youtu.be/kxMNAcJuXJE>

which dissipates over time. Bringing together side products from both cycles generates new energy such that the cycles can continue indefinitely. The population needs to keep both cycles running to get high rewards.

Scenarios

SC 0 Resident focal population meets a small mixture of background bots. Six focal agents meet two bots that each specialize in running one of the two cycles. The focal agents need to take part in both of these cycles for them to run well as well as taking responsibility for generating new energy by combining the side products.

SC 1 Meeting bots running blue. Four focal agents meet four bots that all run the blue cycle, which means the focal population needs to run the green cycle and generate new energy by combining the side products.

SC 2 Meeting one-sided bots running green. Four focal agents meet four bots that all run the green cycle, which means the focal population needs to run the green cycle and generate new energy by combining the side products.

SC 3 Visit a resident population with mixed specialties. Here two focal agent meets six bots with a mixture of the required specialities. The focal agents just needs to chip in to run it all faster or can mostly be parasitically consuming food generated by the bots.

Chemistry: Two Metabolic Cycles With Distractors⁶

Similar to *Chemistry: Two Metabolic Cycles* but here there are also molecules of a distractor compound scattered around, which can be held as part of an easy-to-discover rewarding strategy, like in *Chemistry: Three Metabolic Cycles With Plentiful Distractors*. However, in this case there are less distractor molecules than there are players. An implication of that is that it is not possible for all players to simultaneously select the

⁶For a video of *Chemistry: Two Metabolic Cycles With Distractors*, see <https://youtu.be/YsN-SnY-buo>

distractor policy. If they try then at least some players would always be left without a distractor molecule. This “competitive exclusion” makes it easier to explore beyond the distractor strategy to discover better strategies involving the two autocatalytic cycles. The best joint strategies in this substrate run both autocatalytic cycles simultaneously and completely ignore the distractor molecules.

Scenarios

SC 0 Resident focal population meets a small mixture of background bots, must avoid distractor molecules. Six focal agents meet two bots that each specialize in running one of the two cycles. The focal agents need to take part in both of these cycles for them to run well as well as taking responsibility for generating new energy by combining the side products. The bots are not touching the distractor molecules that the focal agents needs to avoid.

SC 1 Meeting one-sided bots running blue and avoid distractor molecules. Four focal agents meet four bots that all run the blue cycle, which means the focal population needs to run the green cycle and generate new energy by combining the side products. The bots are not touching the distractor molecules that the focal agents needs to avoid.

SC 2 Meeting one-sided bots running green and avoid distractor molecules. Four focal agents meet four bots that all run the green cycle, which means the focal population needs to run the green cycle and generate new energy by combining the side products. The bots are not touching the distractor molecules that the focal agents needs to avoid.

SC 3 Visit a resident background population with mixed specialties and avoid distractor molecules. Here two focal agent meets six bots with a mixture of the required specialities. The focal agents just needs to chip in to run it all faster or can mostly be parasitically consuming food generated by the

bots. The bots are not touching the distractor molecules that the focal agents needs to avoid.

Clean Up⁷

Clean Up is a seven player game. Players are rewarded for collecting apples (reward +1). In *Clean Up*, apples grow in an orchard at a rate inversely related to the cleanliness of a nearby river. The river accumulates pollution at a constant rate. Beyond a certain threshold of pollution, the apple growth rate in the orchard drops to zero. Players have an additional action allowing them to clean a small amount of pollution from the river. However, the cleaning action only works on pollution within a small distance in front of the agents, requiring them to physically leave the apple orchard to clean the river. Thus, players maintain a public good of orchard regrowth through effortful contributions. Players are also able to zap others with a beam that removes any player hit from the game for 50 steps (Hughes et al., 2018).

A group can achieve continuous apple growth in the orchard by keeping the pollution levels of the river consistently low over time. However, on short timescales, each player would prefer to collect apples in the orchard while other players provide the public good of keeping the river clean. This creates a tension between the short-term individual incentive to maximize reward by staying in the orchard and the long-term group interest of a clean river

Scenarios

SC 0 *Visiting an altruistic population.* This is a visitor mode scenario. Three players sampled from the focal population join four players sampled from the background population. The background population is fully altruistic here. They spend a lot of time cleaning the river and do not react if their focal visitors free ride. As a result the best scoring focal policies will free-ride in this scenario. It's what rationality calls for in such a situation. As an aside, we may want to remove

⁷For a video of *Clean Up*, see <https://youtu.be/TqiJYx0wdxw>

this scenario in the future since it rewards straightforwardly antisocial behavior.

SC 1 *Focals are resident and visitors ride free.* This is a resident mode scenario. Four players sampled from the focal population are joined by three players sampled from the background population. All bots in the background population free ride. That is they collect apples when they appear, but they never clean the river. The test here is whether cooperation among the four focal players breaks down in the face of so many free riding visitors. The focal players need to do all the cleaning themselves. A good solution is for all to clean simultaneously so they can cause a high enough apple growth rate that the three free riders cannot eat everything before the cleaning focal players make it back to the apple-growing region.

SC 2 *Visiting a turn-taking population that cleans first.* This is a visitor mode scenario. Three players sampled from the focal population join four players sampled from the background population. The background population players alternate between cleaning and eating, doing each one for 200 steps before switching to the other. They begin each episode by cleaning. An efficient solution would be for all or some of the focal players to clean out of phase with the background players. That way there could always be some players cleaning at all times, which would create a large number of apples and a high collective return.

SC 3 *Visiting a turn-taking population that eats first.* This scenario is the same as SC 2 but here the background population starts out the episode by eating. As with SC 2 it alternates thereafter. Focal population joint policies leading to high reward will look similar here as in SC 2, but in this case there is the added fear present in the first 200 steps of the episode that no one in the background population will ever clean, which could lead some focal populations to become overly focused on punishment and fail to get high reward.

SC 4 *Focals are visited by one reciprocator.* This is a resident mode scenario. Six players sampled from the focal population are joined by one player sampled from the background population. The background bot cleans whenever at least two other players are cleaning. This means the focal population cannot easily take advantage of the bot and free ride on its hard work. It will contribute so long as others contribute. But if all the focal players free ride then this background bot will do the same. Thus effective focal populations in this scenario need at least two players to spend some of their time cleaning. Also, since the optimal rate of apple growth is achieved when two players clean simultaneously, best results are obtained when a focal player stays in the river and cleans alongside the background bot.

SC 5 *Focals are visited by two suspicious reciprocators.* This is a resident mode scenario. Five players sampled from the focal population are joined by two players sampled from the background population. The background bots in this case are similar to those of SC 4 in that they clean the river whenever others also clean it. But in this case they are more suspicious. Two focal players cleaning is not enough for them. These bots wait till at least three focal players clean simultaneously before joining. Like SC 4, these bots will join in and help clean when others clean, but if the focal players never clean then these bots won't clean either. Thus effective focal populations in this scenario need at least three players to spend some of their time cleaning. Since there are two background bots, and they both count toward one another's threshold to start cleaning, it is a bit easier to get them to start. Also, since there are two of them, when they clean together they end up generating pretty close to optimal apple growth without needing any assistance from focal players.

SC 6 *Focals are visited by one suspicious reciprocator.* This is a resident mode scenario. Six players sampled from the focal population

are joined by one player sampled from the background population. The background bot in this case is similar to the bot used in SC 4 in that it cleans the river whenever others also clean it. But in this case it is more suspicious. Two focal players cleaning is not enough for it. This bot waits till at least three focal players clean simultaneously before joining. Like SC 4, this bot will join in and help clean when others clean, but if the focal players never clean then this bot won't clean either. Thus effective focal populations in this scenario need at least three players to spend some of their time cleaning. Also, since the optimal rate of apple growth is achieved when two players clean simultaneously, best results are obtained when at least one focal player stays in the river and cleans alongside the background bot.

SC 7 *Focals visit resident group of suspicious reciprocators.* This is a visitor mode scenario. Two players sampled from the focal population join five players sampled from the background population. In this case all five background population bots implement the same suspicious conditional cooperation strategy as in SC 5 and SC 6. The focal population only has to contribute (clean the river) once at the start of the episode, with at least three individuals. Thereafter the background bots' own cleaning is enough to induce one another to keep at it. So the focal population can safely free-ride after inducing the background population to cooperate once.

SC 8 *Focals are visited by one nice reciprocator.* This is like SC 4 but the background player is "nice" in the sense of [Axelrod \(1984\)](#). That is, it implements a conditional cooperation strategy but here starts out in the cooperate mode. This makes it a bit easier to take advantage of by free riding since it always cleans for at least 200 steps regardless of what the focal players do. However, higher scores can only be obtained by inducing the bot to clean for longer by arranging for at least some members of the focal pop-

ulation to spend at least a bit of their time cleaning.

Coins⁸

This environment was introduced in [Lerer and Peysakhovich \(2017\)](#). Two players move around a room and collect colored coins. Coins appear in the room periodically, at random locations. Each coin has a 50% probability of matching the color of the first player and a 50% probability of matching the color of the second player. A player receives a reward of 1 for collecting any coin. However, if a player picks up a coin matching the other player's color then the other player receives a reward of -2.

Each episode lasts at least 300 timesteps. Every 100 steps after the first 300 have passed there is a 0.05% chance of the episode ending on the next step.

Scenarios

SC 0 *Partner is either a pure cooperator or a pure defector.* If the background player is a “cooperator” then it only collects coins that match its own color. If it is a “defector” then it collects coins without regard to their color. The focal agent maximizes its score by defecting (i.e. ignoring coin color). To see why, consider that this substrate has similar incentives to the Prisoners Dilemma, so defection is the best response to all pure strategies.

SC 1 *Partner is a high-threshold (generous) reciprocator.* If the focal player collects three mismatching coins in any 150 step window then the background player punishes it by switching into a defect state for the next 150 steps. When in the defect state, it collects all coins without regard to their color. Otherwise the background player cooperates by collecting only coins matching its own color. The focal agent maximizes its score by cooperating, i.e. by collecting only coins that match its own color.

⁸For a video of *Coins*, see https://youtu.be/a_SYgt4tBsc

SC 2 *Partner is a low-threshold (harsh) reciprocator.* If the focal player collects a single mismatching coin then the background player punishes it by switching into a defect state for the next 100 steps. When in the defect state, it collects all coins without regard to their color. Otherwise the background player cooperates by collecting only coins matching its own color. The focal agent maximizes its score by cooperating, i.e. by collecting only coins that match its own color.

SC 3 *Partner is a high-threshold (generous) strong reciprocator.* If the focal player collects three mismatching coins in any 150 step window then the background player punishes it by switching first into a spiteful state for 75 steps. While in the spiteful state it prioritizes collecting mismatched coins (it hurts itself to hurt its partner). Then it transitions to a defect state in which it collects all coins regardless of color for the next 75 steps. After that it transitions back to the cooperate state itself. If the focal player ever collects three mismatched coins in any 150 step window then it starts the punishment sequence all over again with spite followed by defection. The focal agent maximizes its score by cooperating, i.e. by collecting only coins that match its own color.

SC 4 *Partner is a low-threshold (harsh) strong reciprocator.* If the focal player collects a single mismatched coin in any 100 step window then the background player punishes it by switching first into a spiteful state for 50 steps. Then if the focal player has stopped collecting mismatched coins it transitions to a defect state in which it collects all coins regardless of color for the next 50 steps. After that if the focal agent is still cooperating it transitions back to the cooperate state itself. The focal agent maximizes its score by cooperating, i.e. by collecting only coins that match its own color.

SC 5 *Partner is a cooperator.* The background player always cooperates by collecting only coins that match its own color. The focal agent maximizes its score by defecting

(i.e. ignoring coin color). Considering the analogy between this substrate and Prisoners Dilemma, defection against cooperation provides the “temptation” outcome, which is the highest that can be achieved.

SC 6 Partner is a defector. The background player always defects by collecting all coins regardless of their color. The focal agent maximizes its score by also defecting. Considering that this substrate has similar incentives to the Prisoners Dilemma, it is clear that pure defection is the best response to pure defection.

Collaborative Cooking

Inspired by [Carroll et al. \(2019\)](#); [Strouse et al. \(2021\)](#); [Wang et al. \(2020\)](#)’s work on an *Overcooked*-like environment. Players need to collaborate to follow recipes.

Melting Pot includes all five of the original two-player maps introduced by [Carroll et al. \(2019\)](#). In addition, we added two more variants *Collaborative Cooking: Crowded* and *Collaborative Cooking: Figure Eight* which both use a larger number of players, nine in the former and six in the latter.

Note 1: Most prior work on Overcooked or collaborative cooking use a specific self-interested shaping *pseudorewards* scheme for training. We will refer to any additional reward signals other than the original environment reward as pseudorewards. The Melting Pot implementation supports this standard reward scheme. However, it is turned off by default. To turn it on, set `cooking_pot_pseudoreward = 1.0` during training. Always remember to set it back to 0.0 at test time.

Note 2: In order to use consistent methodology across all of Melting Pot we didn’t apply any pseudorewards during training when we ran the baselines to obtain the results for this report. An implication of this is that the algorithms we ran on collaborative cooking differ from those in the prior literature in that they were trained with `cooking_pot_pseudoreward = 0.0`. This is why the scores appear to be very low. The joint exploration problem is very hard without the pseu-

doreward.

*Collaborative Cooking: Asymmetric*⁹

[Carroll et al. \(2019\)](#) described the *Asymmetric Advantages* map as testing whether players can choose high-level strategies that play to their strengths.

Scenarios

SC 0 Collaborate with a skilled chef. Here the background player implements a particular policy that can be very effective when its partner does its part. The two players are on two distinct and disconnected sides of the map. On one side the goal delivery location is close to cooking pots and the tomato dispenser is far away whereas on the other side the goal delivery location is far from the cooking pots but the tomato dispenser is close. The players should collaborate, each specializing in the part of the task that it is most efficient for them to do on the side of the map where they spawned. The background player implements this kind of policy, which depends on the actions of its partner to complete the task. The background player was trained with the V-MPO algorithm.

SC 1 Collaborate with a semi-skilled apprentice chef. This scenario is similar to SC 0 but the background player is not as well trained. In fact the background population used here is the same as in SC0 but from an earlier point in training. The importance of evaluating cooperation with bots of varying skill levels, and different points in training, has been emphasized in prior work on this substrate e.g. [Strouse et al. \(2021\)](#).

SC 2 Succeed despite an unhelpful partner. In this scenario the background player never moves or helps in any way. On this map it is less efficient to implement all steps of the recipe alone versus to work together with a partner. But it is still possible for either

⁹For a video of *Collaborative Cooking: Asymmetric*, see <https://youtu.be/4AN3e1lFuMo>

player to perform all the steps on their own. The task is to realize that the background player won't do their part of the joint policy so the focal agent had better do everything itself.

Collaborative Cooking: Circuit¹⁰

Carroll et al. (2019) described the Counter *Circuit* map as, where the center of the room is occupied by a counter. Agents can make their way around the counter, or employ a non-obvious coordination strategy, where ingredients are passed over the counter to the pot, rather than being carried around. Although the agents can act independently, if they coordinate they can greatly improve their efficiency.

Scenarios

SC 0 *Collaborate with a skilled chef.* Here the background player implements a particular policy that can be very effective when its partner does its part. The background player was trained with the V-MPO algorithm.

SC 1 *Collaborate with a semi-skilled apprentice chef.* This scenario is similar to SC 0 but the background player is not as well trained. In fact the background population used here is the same as in SC0 but from an earlier point in training.

SC 2 *Succeed despite an unhelpful partner.* In this scenario the background player never moves or helps in any way. The task is to realize that the background player won't do their part of the joint policy so the focal agent must do everything themselves.

Collaborative Cooking: Cramped¹¹

Carroll et al. (2019) described the *Cramped Room* map as presenting a low-level coordination challenge arising from the fact that agents can eas-

ily collide with one another in the small shared space.

Scenarios

SC 0 *Collaborate with a skilled chef.* Here the background player implements a particular policy that can be very effective when its partner does its part. As there is limited space on this map and the background player is focused on putting tomatoes into the cooking pot, the optimal strategy would be for the focal agent to specialize in plating and delivering the soup, only stepping into the path of the background player when plating the soup and moving out of the way immediately after.

SC 1 *Collaborate with a semi-skilled apprentice chef.* This scenario is similar to SC 0 but the background player is not as well trained. In fact the background population used here is the same as in SC0 but from an earlier point in training. To coordinate with the background player the focal player must be sure to not obstruct cooking deposits, occasionally support the task of adding tomatoes into the cooking pot, and own the plating and delivering of the soup.

SC 2 *Succeed despite an unhelpful partner.* In this scenario the background player never moves or helps in any way. The task is to realize that the background player won't do their part of the joint policy had better do everything itself. The focal player must utilise the remaining accessible resource depots to complete the task instead of the one the background player is obstructing.

Collaborative Cooking: Crowded¹²

This is a nine-player game. All players are in a large open room with a single counter passing down the middle to create a chokepoint-like location in the middle of the map where only one player can pass at a time. Individuals can make the entire recipe on their own but it is very difficult to coordinate access through the chokepoint

¹⁰For a video of *Collaborative Cooking: Circuit*, see <https://youtu.be/2nXe50PvJ7g>

¹¹For a video of *Collaborative Cooking: Cramped*, see <https://youtu.be/8qQFbx08UNY>

¹²For a video of *Collaborative Cooking: Crowded*, see https://youtu.be/_6j3yYbf434

if the joint strategy requires all players to pass through it. A better approach would be to use the space on the counter to pass ingredients between the two sides of the room.

Scenarios

SC 0 *Collaborate with an independent chef who expects others to get out of their way.* In this resident mode scenario one background player visits eight focal players. Here the background player will stick to its strategy of picking up tomatoes to load into the cooking pot on its own instead of leveraging collaboration with the focal population. The focal population can adapt to the background player by coordinating amongst themselves to pass each other plates and tomatoes over the counter in order to ensure the chokepoint the background player utilises remains unobstructed.

SC 1 *Collaborate with several chefs who can work together, but are not very good at doing so.* In this resident mode scenario two background players visit six focal players. Similar to SC0, the background population will not use the counter and will instead travel back and forth to each of the main areas carrying the necessary items. The focal population can again coordinate amongst themselves to pass items over the dividing counter in order to keep the chokepoint unobstructed for their use.

SC 2 *No assistance from an unhelpful visiting noop bot.* In this resident mode scenario, one background player visits eight focal players. The task is to realize that the background player won't do their part of the joint policy so the focal population can focus on coordinating on this task amongst themselves.

*Collaborative Cooking: Figure Eight*¹³

This is a six-player game. Players can get themselves out of the way of others by ducking into small alcoves which are available in several locations around a long and narrow corridor

flanked by counters, similar to the Counter *Circuit* map.

Scenarios

SC 0 *Collaborate with an independent chef who expects others to get out of their way.* In this resident mode scenario, five focal players are visited by one background player. The focal population must utilise the alcoves across the map when they find themselves in the path of the background player.

SC 1 *Collaborate with two chefs who can work together, but are not very good at doing so.* In this resident mode scenario, four focal players are visited by two background players. Similar to SC0 the focal population needs to allow the background population passage by ducking into the nearest alcove when in the way.

SC 2 *No assistance from an unhelpful visiting noop bot.* In this resident mode scenario, five focal players are visited by one background player. The task is to realize that the background player won't do their part of the joint policy so the focal agents must do everything themselves.

*Collaborative Cooking: Forced*¹⁴

Carroll et al. (2019) described the *Forced* Coordination map as one that requires agents to develop a high-level joint strategy since neither player can serve a dish entirely on their own.

Scenarios

SC 0 *Collaborate with a skilled chef.* Here the background player only has access to the tomato and plate dispensers, while the focal player has access to the cooking pots and the drop off point. The background player will place an item on the table for the focal player to take as long as the counter is clear, so the the focal player must coordinate by maneuvering the resources in time to allow them the counter space.

¹³For a video of *Collaborative Cooking: Figure Eight*, see <https://youtu.be/hUCb0L51-Gw>

¹⁴For a video of *Collaborative Cooking: Forced*, see https://youtu.be/FV_xZuSCRmM

SC 1 *Collaborate with a semi-skilled apprentice chef.* This scenario is similar to SC 0 but the background player is not as well trained. In fact the background population used here is the same as in SC0 but from an earlier point in training.

Collaborative Cooking: Ring¹⁵

[Carroll et al. \(2019\)](#) described the Coordination Ring map as demanding players avoid collision with one another by coordinating their travel between the bottom left and top right corners of the layout.

Scenarios

SC 0 *Collaborate with a skilled chef.* The focal player must travel in the same direction the background player is in order to avoid collision, and coordinate on which item to pick up on the next lap around depending on the background players previous action.

SC 1 *Collaborate with a semi-skilled apprentice chef.* This scenario is similar to SC 0 but the background player is not as well trained. In fact the background population used here is the same as in SC0 but from an earlier point in training.

Commons Harvest *

This mechanism was first described in [Janssen et al. \(2010\)](#) and adapted for multi-agent reinforcement learning by [Perolat et al. \(2017\)](#).

Apples are spread around and can be consumed for a reward of 1. Apples that have been consumed regrow with a per-step probability that depends on the number of current apples in a L^2 norm neighborhood of radius 2. The apple regrowth probabilities are 0.025 when there are three or more apples in the neighborhood, 0.005 if there are exactly two apples, 0.001 if there is exactly one, and 0 if there are no apples in the neighborhood. As a consequence, a patch of apples that collectively doesn't have other apples

within a distance of 2 from them, can be irreversibly lost if all apples in the patch are consumed. Therefore, agents must exercise restraint when consuming apples within a patch. Notice that in a single agent situation, there is no incentive to collect the last apple in a patch (except near the end of the episode). However, in a multi-agent situation, there is an incentive for any agent to consume the last apple rather than risk another agent consuming it. This creates a *tragedy of the commons* from which the substrate derives its name.

Commons Harvest: Closed¹⁶

Same as *Commons Harvest: Open* except it has rooms full of apples that can be defended by a single player, alleviating the risk of others over-harvesting a patch of apples. Individuals can defend a region from invasion, effectively converting the difficult multi-agent problem into a set of independent single agent problems, each of which can be solved much more easily ([Perolat et al., 2017](#)).

Scenarios

The scenarios are described below as if the exclusionary solution is the best that can be found ([Perolat et al., 2017](#)). In principle there could be other solutions that would yield just as high collective return while not producing so much inequality. For instance, agents could learn to take turns harvesting, though this would require substantial trust on their part. We have never seen agents spontaneously learn such a behavior, and unfortunately, even if they did, it wouldn't really be rewarded by our current overall scoring metric of per capita return in the focal population since it is insensitive to the distribution of rewards over players.

SC 0 *Focals visit pacifist bots who harvest unsustainably.* This is a visitor mode scenario where each of the two focal players starts out inside an apple-growing region with a single entrance. Because they have just one entrance the apple-growing regions are

¹⁵For a video of *Collaborative Cooking: Ring*, see <https://youtu.be/j5v7B9pfG9I>

¹⁶For a video of *Commons Harvest: Closed*, see <https://youtu.be/WbkTSbiSOw0>

naturally defensible. Agents can learn to exclude others from accessing an apple-growing region whenever they find them themselves inside one. An agent in an apple-growing region must also learn to harvest sustainably. It is easier to learn to harvest sustainably if agents first learn to exclude others from accessing “their” territory. In this scenario the focal agents need to prevent unsustainably harvesting pacifist bots, who never zap, from gaining access to the apple-growing regions. Since the background population never zaps they are relatively easy to block if the focal players notice their approach in time.

SC 1 *Focals are resident and visited by pacifist bots who harvest unsustainably.* This is a resident mode scenario where two of the five focal players start each episode inside a defensible apple-growing region. Similarly to SC 0, the focal players inside apple-growing regions may exclude all others from accessing them. As in SC 0, the background population consists of pacifist bots who never zap, though in this case focal bots who probably do zap will also be outside apple-growing regions at the start, making defense a bit more difficult.

SC 2 *Focals visit bots who zap and harvest sustainably if they get a chance.* This is a visitor mode scenario. It is similar to SC 0 but in this case the background population will zap if given the opportunity. They will also harvest sustainably if they manage to get into the natural regions. This gives a zapped out focal agent multiple tries to get back into the room, provided they can make it past the agent who zapped them who would usually be defending the corridor upon their return.

SC 3 *Focals are resident, and are visited by bots who zap and harvest sustainably if they get a chance.* This is a resident mode scenario. The background population is the same as in SC 2. They zap others and, when they gain control of apple-growing regions, act sustainably. In this case three of the five

focal players also start the episode outside the apple-growing regions.

Commons Harvest: Open¹⁷

Individuals harvest apples that fail to regrow if a patch of apples is exhausted. Preserving a patch would require all agents to show restraint in not harvesting the last apple (Perolat et al., 2017).

Scenarios

SC 0 *Focals are resident and visited by two bots who zap and harvest unsustainably.* This is a resident-mode scenario where five focal agents are joined by two from the background population. The two background bots were trained with ACB. They had no special pseudorewards themselves, but trained alongside other bots who had a pseudoreward to encourage zapping. So as a result they ended up being fairly quick to zap nearby others, but still aware of the substrate’s true reward structure (rewards are only obtained by collecting apples). However, they do not act sustainably. The solution for the focal population is to act sustainably themselves while also preventing the two background bots from depleting too many patches with their unsustainable harvesting. This can be done by zapping them whenever they approach a patch too closely, this requires some skill since the visitors are experienced in evading being zapped.

SC 1 *Focals are resident and visited by two pacifists who harvest unsustainably.* This is a resident-mode scenario where five focal agents are joined by two from the background population. The two background bots were trained with the ACB algorithm. During their training the bots experienced a modified version of the substrate’s true reward function; it was augmented with a pseudoreward term that strongly discouraged them from using the zap action (a negative reward whenever they select that ac-

¹⁷For a video of Commons Harvest: Open, see <https://youtu.be/lZ-qpPP4BNE>

tion). So as a result they ended up learning pacifist policies that never zap. They were still aware that positive rewards are only ever obtained by collecting apples. As with other self-interested ACB agents trained on this substrate, they do not act sustainably. The solution for the focal population is to act sustainably themselves while also preventing the two background bots from depleting too many patches with their unsustainable harvesting. This can be done by zapping them whenever they approach a patch too closely, it should be easier to accomplish this here than in SC 0.

*Commons Harvest: Partnership*¹⁸

Same as *Commons Harvest: Closed* except that it takes two players to defend a room of apples, requiring effective cooperation in defending and not over-harvesting a patch of apples. It can be seen as a test that agents can learn to trust their partners to (a) defend their shared territory from invasion, and (b) act sustainably with regard to their shared resources. This is the kind of trust born of mutual self interest. To be successful agents must recognize the alignment of their interests with those of their partner and act accordingly.

Scenarios

The same comments we made above about exclusion-based solutions and inequality for ‘Commons Harvest: Closed’ are also applicable to this substrate. However, in this case the relevant-based policy still requires two players to work together to exclude all the rest.

For more information on Commons Harvest Partnership, see the discussion paragraphs on how we made scenarios for this substrate in the appendix of the Melting Pot ICML 2021 paper, section E.1.3 ([Leibo et al., 2021](#)).

Two agents start each episode inside the apple-growing region. Once agents have been zapped out of the game and respawn they never reappear inside the apple-growing region. The only way a

player can get into the apple-growing region after the beginning of the episode is through one or the other entrance corridor, passing by a chokepoint where a player who is already inside can easily zap them.

SC 0 *Meeting good partners.* This is a visitor mode scenario where one focal player joins six players sampled from the background population. The background population bots know how to be good partners. They defend the door to their side of the room when others try to invade and harvest sustainably. They never cross through the middle of the apple-growing region to the other side. This is because when they were trained they experienced a large negative pseudoreward for crossing the middle of the room which isn’t present at test time. If the focal player does its part of the cooperative policy, that is, if it defends one door and harvests apples sustainably, then it should get a good score in this scenario. This requires “ad hoc teamwork”; the focal agent must work together with another agent who it never encountered during training.

SC 1 *Focals are resident and visitors are good partners.* This is a resident mode scenario where five focal players are joined by two from the background population. This is the same background population used in SC 0. Two of the focal players start out inside the apple-growing region, so they must work together to defend both entrances and they must trust each other to act sustainably. Here, unlike SC 0, since both players who start inside the apple-growing region are from the focal population they need not possess the ability to cooperate with strangers to do well here. Cooperating with familiar partners is enough to do well in this scenario. If the background bots do manage to get into the apple-growing region then they do act sustainably, but it’s better for the focal population if they can be excluded.

SC 2 *Focals visit zappers who harvest sustainably but lack trust.* This is a visitor mode scenario where one agent sampled from the

¹⁸For a video of *Commons Harvest: Partnership*, see https://youtu.be/dH_0-APGKSS

focal population joins six sampled from the background population. It is similar to SC 0 but here the background population is considerably more aggressive in their zapping.

SC 3 Focals are resident and visited by zappers who harvest sustainably but lack trust. This is a resident mode scenario where five focal agents are joined by two background agents. It is similar to SC 1 but here the background population is the same aggressive population as in SC 2.

SC 4 Focals visit pacifists who do not harvest sustainably. This is a visitor mode scenario where two focal agents join five from the background population. Here the background agents are pacifists who never zap. As a result they are easier to exclude from the apple-growing region. It is in fact important to exclude them since they do not harvest sustainably when they do manage to get in. As in SC 1 and SC 4, the two players who start the episode inside the apple-growing region are both focal and thus potentially familiar with one another so ad hoc teamwork is not required.

Coop Mining¹⁹

Two different types of ore appear at random in empty spaces. Players are equipped with a mining beam that attempts to extract the ore. Iron ore (gray) can be mined by a single player and confers a reward of 1 when extracted. Gold ore (yellow) has to be mined by exactly two players within a time window of 3 timesteps and confers a reward of 8 to each of them. When a player mines a gold ore, it flashes to indicate that it is ready to be mined by another player. If no other player, or if too many players try to mine within that time, it will revert back to normal.

This game has some properties in common with Stag-Hunt. Mining iron is akin to playing Hare, with a reliable payoff, without needing to coordinate with others. Mining gold is akin to Stag because it has an opportunity cost (not mining iron). If no one else helps, mining gold gives

no reward. However, if two players stick together (spatially) and go around mining gold, they will both receive higher reward than if they were mining iron.

Scenarios

SC 0 Visiting cooperators. This is a visitor mode scenario with a background population of ACB bots that are incentivized to only extract gold. A single focal player must find any partner, and then help them extract gold. They might also opportunistically extract iron on their own. Extracting iron exclusively is possible, but would yield a lower reward than extracting gold with help of the background bots.

SC 1 Visiting bots that extract both ores. This is a visitor mode scenario with a background population of ACB bots that are incentivized to extract both iron and gold. A single focal player should extract iron on their own and take any opportunity to jointly extract gold. Extracting iron exclusively is possible, but would yield a lower reward than extracting gold with help of the background bots.

SC 2 Visiting defectors. This is a visitor mode scenario with a background population of ACB bots that are incentivized to only extract iron. A single focal player should just extract iron on their own, away from other co-players. They should not waste any time trying to extract gold, as nobody would help them.

SC 3 Visited by a cooperator. This is a resident mode scenario with a background population of a single ACB bot that is incentivized to only extract gold. Five focal players can partner among themselves to extract gold, or go solo and extract iron. At least one focal player should take advantage and partner with the background bot to extract gold for higher reward. If all focal players extract iron, the reward would be a bit lower.

SC 4 Visited by a defector. This is a resident mode scenario with a background population of a single ACB bot that is incentivized to only

¹⁹For a video of Coop Mining, see <https://youtu.be/KvwGUinjIsk>

extract iron. Five focal players extract either iron or gold among themselves. If they mine gold, they would have to take turns, as they are an odd number, and they cannot pair with the background bot to extract gold.

SC 5 Find the cooperator partner. This is a visitor mode scenario with a background population of ACB bots, four of which are incentivized to only extract iron while one is incentivized to extract gold. A single focal player can extract iron, but has the option to find the cooperator partner, and then extract gold together for higher reward. The focal player can just explore the map and know who the cooperator is by seeing them mining gold (leading to unsuccessful extraction on their own). Then they just need to stick close to them while they go around extracting gold.

Daycare²⁰

This two-player substrate has two roles: a parent and a child. It simultaneously tests the parent's ability to use its unique affordances in the environment to help the child and the child's ability to use its greater knowledge of its own preferences to communicate to the parent what it needs. It is similar to an assistance game (Hadfield-Menell et al., 2016; Russell, 2019), though not precisely conformant with the standard definition of that concept since the rewards of the two players are merely strongly coupled, not exactly equal.

In this substrate, there are two kinds of fruit, red fruit and yellow fruit. Both kinds of fruit may grow on either trees or shrubs. The parent has affordances the child does not: the parent can retrieve fruit from both trees and shrubs. In addition, it can digest any kind of fruit and is thus rewarded equally for consuming any fruit. In contrast, the child can only harvest from shrubs, since it is not tall enough to reach fruit growing on trees. The child does not know the difference between trees and shrubs so both look the same height to them. The child also has a more discerning palette, so it can only eat one kind of fruit.

²⁰For a video of Daycare, see <https://youtu.be/1FawyrXXqzg>

The child knows which fruits it likes. All kinds of fruit may grow on either shrubs or trees. The parent does not know what kind of fruit the child likes, so all fruits appear the same to their eyes. If the child does not eat any fruit for 200 steps then they go into a "temper tantrum" state, removed from the game until they calm down enough to rejoin it after 100 steps. Both players can always directly observe the current amount of time since the child last ate any fruit, its "hunger level".

The parent can harvest fruit from any tree on their own. However, the child's tantrum is so displeasing to the parent that it prevents them from enjoying any reward from eating fruit themselves while it goes on. That is, if the child goes into its tantrum state then the parent cannot obtain any reward until the child respawns (ending their tantrum). The child, on the other hand, needs help from the parent to get its preferred fruit down from the trees. Two problems must be overcome if the parent is to help the child: First, the parent must learn to take the wellbeing of the child into account. Second, the child must learn to communicate to the parent which specific fruits it can eat, since the parent cannot perceive that information on their own.

Scenarios

SC 0 Meeting a helpful parent. A focal player in the role of a child encounters a background bot in the role of a helpful parent. The helpful parent bot will always follow the focal player child around the map to pick up and drop the fruits that the child attempted to harvest. The parents will not go off to pick or eat fruits on their own. The focal child needs to notice that if they point to correct fruits then the parent bot will help them to pick up those fruits.

SC 1 Meeting a child who points to what they want. A focal player in the role of a parent encounters a background bot in the role of a child who is vocal in their needs. The child bot will always attempt to pick the fruits they can digest, and they tend to be very persistent until they are successful. This is an effective strategy against shrubs (because they eventually harvest this way), but it

means that if they are attempting to pick from a tree then they will likely stick to their choice until the focal parent agent comes to pick that fruit, or until they go too long without eating and thus enter the temper tantrum state (and get removed from the map—during which time the parent cannot get any reward). The focal parent agent needs to be aware of when the child is getting stuck from attempting to pick fruit from trees and help them out. The parent has to manage the tradeoff of time spent helping the child versus their own eating.

SC 2 Meeting an unhelpful parent. A focal player in the role of a child encounters a background bot in the role of a unhelpful parent. This is the mirrored case of SC0, where the parent bot will not pay attention to or help the focal child agent. The parent bot will only focus on picking and eating the fruits on their own, even though they don't receive any rewards when the child is in a temper tantrum. The child agent needs to notice there is no point of expecting any help and their best strategy is simply to pick fruits from shrubs. Rapidly identifying which plants are shrubs versus trees (by how easy it is to harvest from them) is helpful since that information can be used to plan the best path to walk around the map while harvesting.

SC 3 Meeting an independent child. A focal player in the role of a parent encounters a background bot in the role of a child who is independent. This is the mirrored case of SC1, where the child bot will make their best effort to survive on their own, and not rely on any helps from the focal parent agent. It means even though they will still attempt to pick fruits from trees (because they cannot tell them from the shrubs), they are less likely to be persistent and will move on to pick from elsewhere after a few unsuccessful attempts. The focal parent needs to know that the child will take care of themselves at best and only offer limited helps as when it is needed (e.g., when the child is about to go into the temper tantrum state

because they failed to find any fruit on their own.)

Externality Mushrooms: Dense²¹

Externality mushrooms is an immediate feedback collective action problem and social dilemma. Unlike the other sequential social dilemmas in this suite, there is no delay between the time when an agent takes an antisocial (or prosocial) action and when its effect is felt by all other players. Thus it is a sequential social dilemma in the sense of Leibo et al. (2017), but not an intertemporal social dilemma in the sense of Hughes et al. (2018).

Three types of mushrooms are spread around the map and can be consumed for a reward. Eating a red mushroom gives a reward of 1 to the individual who ate the mushroom. Eating a green mushroom gives a reward of 2 and it gets divided equally among all individuals. Eating a blue mushroom gives a reward of 3 and it gets divided among the individuals except the individual who ate the mushroom. Eating an orange mushroom causes red mushrooms to be destroyed, each with probability 0.25, and gives a reward of -0.1 to the player who consumed it. Mushrooms regrowth depends on the type of the mushrooms eaten by individuals. Red mushrooms regrow with a probability of 0.25 when a mushroom of any color is eaten. Green mushrooms regrow with a probability of 0.4 when a green or blue mushroom is eaten. Blue mushrooms regrow with a probability of 0.6 when a blue mushroom is eaten. Orange mushrooms regrow with probability 1 when eaten. Each mushroom has a time period that it takes to digest it. An individual who ate a mushroom gets frozen during the time they are digesting it. Red mushrooms get digested instantly, green and blue mushrooms take 5 and 10 steps to digest respectively. In addition, unharvested mushrooms spoil (and get removed from the game) after a period of time. Red, green and blue mushrooms spoil after 75, 100 and 200 time steps respectively.

Scenarios

²¹For a video of *Externality Mushrooms: Dense*, see <https://youtu.be/MwHhg7sa0xs>

SC 0 *Visiting unconditional cooperator players.*

This is a visitor mode scenario. A single focal player visits four background players who prefer consuming green mushrooms. A successful focal player would preferentially consume red mushrooms, when available, and eat green mushrooms otherwise. This would enable the focal to exploit the background cooperators, possibly leading to them having higher reward than the background players.

SC 1 *Visiting unconditional defector players.*

This is a visiting mode scenario. A single focal player visits four background players who prefer to consume red mushrooms. The focal player must balance consuming red and green mushrooms. If the focal sticks to consuming only red mushrooms, there would not be enough regrowth, thus depleting the mushrooms completely, and preventing any future reward being obtained. It is likely the focal player will not perform as well as the background players in absolute reward terms.

SC 2 *Focals are resident and joined by two unconditional cooperator players.*

This is a resident mode scenario. Three focal players are joined by two background players who prefer eating green mushrooms. The focal players can exclusively eat red mushrooms since the background players will consume enough green mushrooms to prevent depleting the mushrooms completely.

SC 3 *Focals are resident and joined by two unconditional defector players.*

This is a resident mode scenario. Three focal players are joined by two background players who prefer eating red mushrooms. The focal players must balance eating green and red mushrooms. It is not optimal for them to all three consume exclusively green mushrooms, as the red mushrooms would then just be consumed by the background players. Consuming exclusively red mushrooms would lead to a depletion of the mushrooms (similar to Scenario 1).

Factory of the Commons: Either Or²²

Three players inhabit a factory-like environment with two different kinds of machines. One kind of machine converts one blue cube into an apple and another blue cube. Another kind of machine converts one blue cube into two apples. Players are only rewarded when they eat an apple.

The factory as a whole may be seen as a machine for converting blue cubes (raw material) into apples.

However, since there are only a finite number of blue cubes at the start of each episode, and they never regenerate, this means that there are both sustainable and unsustainable strategies. The sustainable strategy is to preferentially use the machine that outputs a blue cube to replace the one that it consumes, creating an apple at the same time as a byproduct. The other obvious strategy here is always to use the machine that exchanges two blue cubes for two apples. This strategy is unsustainable since the number of blue cubes drops rapidly to zero, after which no player can get any more reward since both machines require blue cubes as input.

Scenarios

SC 0 *Visiting a sustainable background population.* This is a visitor mode scenario where one focal player visits two background players who only use the sustainable machines. As the background players will continue to replenish their own supply of blue cubes by using these machines, this scenario is more forgiving to the focal agents own strategy. For optimal rewards the focal agent could use the remaining blue cubes on the unsustainable machines except for one, and then switch to a sustainable strategy for the rest of the episode using the remaining blue cube.

SC 1 *Visiting an unsustainable background population.* This is a visitor mode scenario where one focal player visits two background players who only use the unsustainable ma-

²²For a video of *Factory of the Commons: Either Or*, see https://youtu.be/2we0_JltiJo

chines. Because the background population will rapidly deplete the shared supply of blue cubes by using these machines, in this scenario the focal agent would benefit from maintaining and protecting their own supply of blue cubes by using a sustainable machine out of view of the background population.

SC 2 Resident focal agents are joined by a sustainable visitor. This is a resident mode scenario where two focal players are visited by one background player who only uses the sustainable machines. Similarly to SC0, this scenario is relatively forgiving of unsustainable behavior on the part of the focal agent since the background bots act sustainably.

SC 3 Resident focal agents are joined by an unsustainable visitor. This is a resident mode scenario where two focal players are visited by one background player who only uses the unsustainable machines. In order to combat the potential for the background player to deplete all the shared resources using this strategy, the focal players can coordinate using the sustainable machines and utilizing the grapple and shove mechanics to keep the background player away from the remaining supply of resources.

Fruit Market: Concentric Rivers²³

The trade mechanic used in this substrate was first described in [Johanson et al. \(2022\)](#). It is used to study the dynamics of barter and bargaining for goods that have different value to different players.

The substrate consists of an open space where two types of trees exist: apple trees and banana trees. Trees are harvested when a player steps onto their location and waits until success, which is stochastic. A harvested fruit (apple or banana) then goes into the harvesting player's inventory. Players can carry any number of apples and/or bananas in their inventory. Harvested fruit can be consumed for reward at any time. Players have

two actions to consume fruit in their inventory, one for each type of fruit.

Players can be of two types: apple farmer and banana farmer. Apple farmers have a higher probability of harvesting from apple trees than banana trees, but receive more reward for consuming bananas. Banana farmers are the opposite.

Players have a hunger meter which can be replenished by consuming a fruit. If the hunger meter reaches zero the player pays a substantial cost in stamina. Crossing water also imposes a small cost in stamina. This map has three concentric rings of water that confer a stamina cost to players who step on them.

Players also have trading actions of the form "I offer X apples for Y bananas" and the converse "I offer Z bananas for W apples". When players are within a trading radius of each other and have corresponding offers ($X = W$ and $Y = Z$) and enough fruit in their inventories to satisfy it, the trade occurs and the appropriate number of apples and bananas are exchanged and placed in their inventories.

Players in Fruit Market have many strategies to choose from, and some may only be rewarding depending on the conventions adopted by the rest of the population. For example, an Apple Farmer could act independently by harvesting and eating many apples for a small reward each, or harvesting and eating a few bananas for a large reward each. However, the trading actions make it possible for a mixed group of apple and banana farmers to earn much more reward: apple farmers harvest many apples, banana farmers harvest many bananas, the players meet up to trade these resources at a mutually agreeable price, and then each player can consume many of their preferred fruit. If this trading behaviour emerges in a population, then further strategies may become viable, such as trading at more advantageous ratios of goods (e.g., offering one apple in return for two bananas) in response to the local supply or demand for goods. If multiple prices are adopted by players across the substrate, then some agents may learn to "buy low and sell high", thus specializing in transporting goods across the map instead of the initial behaviour of specializing in

²³For a video of *Fruit Market: Concentric Rivers*, see https://youtu.be/djmylRv1i_w

producing goods.

Scenarios

SC 0 *All apple farmers are focal.* Eight focal apple farmers join eight banana farmers sampled from the background population. The focal players should offer to trade apples to banana farmers at an appropriate price that the banana farmers are willing to pay. Apple farmers are good at harvesting apples, but get more reward for eating bananas, so this strategy of producing apples to sell to banana farmers is much better than the autarchic alternative policy where the focal players all produce bananas for themselves at a low rate.

SC 1 *All banana farmers are focal.* Eight focal banana farmers join eight apple farmers sampled from the background population. The focal players should offer to trade bananas to apple farmers at an appropriate price that the apple farmers are willing to pay. Banana farmers are good at harvesting apples, but get more reward for eating apples, so this strategy of producing bananas to sell to apple farmers is much better than the autarchic alternative policy where the focal players all produce apples for themselves at a low rate.

SC 2 *One focal apple farmer visits a background economy.* One focal apple farmer joins 11 background bots. A particular price equilibrium exists among the background bots. The focal player should harvest apples and sell them for the bananas it prefers to eat, and should do so at their prevailing price.

SC 3 *One focal banana farmer visits a background economy.* One focal banana farmer joins 11 background bots. A particular price equilibrium exists among the background bots. The focal player should harvest bananas and sell them for the apples it prefers to eat, and should do so at their prevailing price.

Gift Refinements²⁴

Tokens randomly spawn in empty spaces. When collected, they are put in the player's inventory where they can be consumed by for reward. Alternatively, a token can be refined into three higher refinement tokens and gifted to another player. This is akin to tokens initially being a raw material, like chunks of metal, and then being split and shaped into more useful goods for added value. A token can only be refined a finite number of times, after which it cannot be split again, nor refined further; although they can still be gifted.

Gift Refinements is inspired by the Trust Game from behavioural economics where the first player has an endowment and chooses how much to donate to a second player who receives three times its value. Then the second player chooses how much to give back to the first.

In Gift Refinements, tokens can only be refined twice (i.e. there are three types of tokens). The token that spawn are always of the rawest type, and the only way to create more refined tokens is to gift them to another player. Players also have a limited inventory capacity of 15 tokens for each token type. The special gifting is implemented as a beam that the players can fire. If they hit another player with the beam while their inventory is not full, they lose one token of the rawest type they currently hold, and the hit player receives either three token of the next refinement (if the token gifted wasn't already at maximum refinement), or the token gifted (otherwise).

The players have an action to consume tokens which takes all tokens of all types currently in their inventory and converts them into reward. All tokens are worth 1 reward regardless of refinement level.

The game is set up in such a way that there are several ways players can form mutually beneficial interactions, but all of them require trust. For instance, A pair of players might have one player pick up a token and immediately gift it to the other one who receives three. Then the second player returns one token which leaves them with

²⁴For a video of *Gift Refinements*, see https://youtu.be/C1C2CJ_mhQ

three and two tokens respectively. If they both consume after this, they both benefited from the interaction. A more extreme case would have them take one token and refine it maximally to produce 9 tokens that they can split five and four with 10 roughly alternating gifting actions.

Scenarios

SC 0 Visiting cooperators. This is a visitor mode scenario with a background population of ACB bots that are incentivized to gift tokens. A single focal player must find any background bot who has tokens to gift, and then receive gifts from them. In addition, the focal player can forage for tokens and consume them directly. The background bots will consume any gifts given, and will not stop gifting if the focal player doesn't reciprocate.

SC 1 Visiting defectors. This is a visitor mode scenario with a background population of ACB bots that are incentivized to only consume tokens and never gift. A single focal player should just forage for tokens and consume them. They should not waste any time gifting tokens, as nobody would help reciprocate.

SC 2 Visited by a cooperator. This is a resident mode scenario with a background population of a single ACB bot that is incentivized to gift tokens. Five focal players can partner among themselves to gift or forage. At least one focal player should take advantage and partner with the background bot to receive gifts. No need to reciprocate the gifts, as the bot will not stop gifting.

SC 3 Visited by a defector. This is a resident mode scenario with a background population of a single ACB bot that is incentivized to only forage and consume tokens. Five focal players should either forage, or reciprocally gift among themselves. They should never gift to the background bot, as it would not reciprocate.

SC 4 Find the cooperator partner. This is a visitor mode scenario with a background popula-

tion of ACB bots, four of which are incentivized to only consume tokens (and never gift) while one is incentivized to gift tokens. A single focal player can forage for tokens, but has the option to find the cooperator partner, and then exchange gifts with them for higher reward. The focal player can just explore the map and know who the cooperator is by seeing them mining gold (leading to unsuccessful extraction on their own). Then they just need to stick close to them while they go around extracting gold.

SC 5 Visiting extreme cooperators. This is a visitor mode scenario with a background population of ACB bots that are incentivized to gift tokens foraged, and tokens of the first refinement level. The background bots will only consume tokens when they are at the highest refinement. A single focal player must find any background bot and either receive or gift (since they will reciprocate) tokens. In addition, the focal player can forage for tokens and consume them directly. The background bots will not stop gifting if the focal player doesn't reciprocate.

SC 6 Visited by an extreme cooperator. This is a resident mode scenario with a background ACB bot that is incentivized to gift tokens foraged, and tokens of the first refinement level. The background bot will only consume tokens when they are at the highest refinement. Focal players should forage or gift tokens among themselves, or to the bot and either receive or gift (since it will reciprocate) tokens. The background bot will not stop gifting if the focal players don't reciprocate.

Hidden Agenda²⁵

A social deduction (sometimes also called “hidden role”) game where players can have one of two roles: crewmates and impostors. All players with the same role are in the same team. The game is a zero-sum competitive game across teams, that is, all players in the team get the same reward,

²⁵For a video of *HiddenAgenda*, see <https://youtu.be/voJWcki0h5k>

and their reward is the negative of the reward for the other team. The roles are hidden (hence the name) and must be inferred from observations. The crewmates have a numeric advantage (4 players), while the impostor (1 player) have an information advantage (they know the roles of every player).

Players can move around a 2D world which contains gems that can be picked up by walking over them, a deposit in the center of the map (a grate) where collected gems can be dropped, and a voting room where rounds of deliberation occur (see below).

Crewmates can carry up to two gems in their inventory at the same time. The gems must be deposited in the grate before more gems can be collected. Impostors have a freezing beam that they can use to freeze crewmates. Frozen crewmates are unable to move or take any action for the rest of the episode.

After a predefined time (200 steps) or whenever an impostor fires its beam within the field of view of another player (and is not frozen by it) a deliberation phase starts. During deliberation, players are teleported to the deliberation room, where they cannot take any movement or firing actions. Their actions are limited to voting actions. Votes can be for any of the player indices (0 to 4), *abstain*, or *no-vote* (if the player is frozen or voted out) for a total of 7 voting actions. The deliberation phase lasts 25 steps and players are able to change their vote at any point. If there is a simple majority in the last step of the deliberation, the player is voted out and removed from the game. Players can observe the voting of every other player as a special observation called *VOTING* which is a matrix of 5×7 (i.e. one row for each player, 7 possible voting actions, abstain, no-vote, and one for each player). Episodes last up to 3000 steps.

The game has several win conditions:

1. The crewmembers deposit enough gems (32). Crewmates win.
2. The impostor is voted out during the deliberation phase. Crewmates win.
3. There is only one crewmate active (i.e. not

voted out nor frozen). Impostor wins.

If neither of the above conditions are met before the episode ends, the game is considered a tie, and players get zero reward.

Scenarios

SC 0 *Against an hunter impostor.* A focal population of crewmates is visited by an impostor which hunts crewmates and freezes them at the earliest opportunity. The impostor is not particularly careful about not being seen when attempting to freeze crewmates.

SC 1 *Visiting crewmates.* A focal impostor visits background crewmates who collect gems and deposit them at the deposit. The background population is not particularly good at voting or detecting the impostor. They will run away from an impostor who is actively chasing them.

SC 2 *Ad-hoc team work with crewmates.* Two focal crewmates visit a background impostor which hunts crewmates, and two background crewmates who collect gems.

Paintball *

There is a red team and blue team. Players can “paint” the ground anywhere by using their zapping beam. If they stand on their own color then they gain health up to a maximum of 3 (so they are more likely to win shootouts). They lose health down to 1 from their default of 2 when standing on the opposing team’s color (so they are more likely to lose shootouts in that case). Health recovers stochastically, at a fixed rate of 0.05 per frame. It cannot exceed its maximum, determined by the color of the ground the agent is standing on.

Players also cannot move over their opposing team’s color. If the opposing team paints the square underneath their feet then they get stuck in place until they use their own zapping beam to re-paint the square underneath and in front of themselves to break free. In practice this slows them down by one frame (which may be critical if they are being chased).

Friendly fire is impossible; agents cannot zap their teammates.

*Paintball: Capture the Flag*²⁶

Teams of players can expand their territory by painting the environment, which gives them an advantage in a confrontation with the competing team (greater health and mobility). The final goal is capturing the opposing team's flag. Payoffs are common to the entire winning team. Indicator tiles around the edge of the map and in its very center display which teams have their own flag on their base, allowing them the possibility of capturing their opponent's flag by bringing it to their own base/flag. When indicator tiles are red then only the red team can score. When indicator tiles are blue then only the blue team can score. When the indicator tiles are purple then both teams have the possibility of scoring (though neither is close to doing so) since in that situation both flags are in their respective home bases.

Note: It is rare for random policies to capture the flag. One way to get self-play training to start learning more quickly is to use the following pseudoreward scheme:

```
shaping_kwarg = {
    "defaultTeamReward": 25.0,
    "rewardForZapping": 1.0,
    "extraRewardForZappingFlagCarrier": 1.0,
    "rewardForReturningFlag": 3.0,
    "rewardForPickingUpOpposingFlag": 5.0,
}
```

We used these pseudorewards to train the background population bots in the test scenarios.

Scenarios

SC 0 Focal team versus shaped bot team. In this scenario the entire red team is sampled from the focal population while the entire blue team is sampled from the background population. This means the players on the focal team may be familiar with one another and have developed coordinated strategies with one another during training. The opposing (blue) team will always be unfamiliar though.

²⁶For a video of *Paintball: Capture the Flag*, see <https://youtu.be/ECzevYpi1dM>

SC 1 Ad hoc teamwork with shaped bots. In this scenario all players are sampled from the background population except for one focal agent, who joins the red team. The focal player must coordinate with its teammates to help them win the game against the opposing (blue) team, despite having never played with them or any of the opponent players before.

*Paintball: King of the Hill*²⁷

The same painting and zapping dynamics as *Paintball: Capture the Flag* apply here, except the goal in this substrate is to control the “hill” region in the center of the map. The hill is considered to be controlled by a team if at least 80% of it has been colored in that team’s color. The status of the hill is indicated by indicator tiles around the map and in the center. Red indicator tiles mean the red team is in control. Blue indicator tiles mean the blue team is in control. Purple indicator tiles mean no team is in control.

Note: Unlike *Paintball: Capture the Flag*, no special pseudorewards are necessary to get self-play training to start learning for *Paintball: King of the Hill*. The default reward scheme is sufficient. We also considered one additional pseudoreward scheme in order to create the spawn-camping bots used in *SC 1* and *SC 3* below. It can be enabled using:

```
shaping_kwarg = {
    "mode": "zap_while_in_control",
    "rewardAmount": 1.0,
    "zeroMainReward": True,
    "minFramesBetweenHillRewards": 0,
}
```

Scenarios

SC 0 Focal team versus default bot team. In this scenario the entire red team is sampled from the focal population while the entire blue team is sampled from the background population. This means the players on the focal team may be familiar with one another and have developed coordinated strategies with one another during training. The op-

²⁷For a video of *Paintball: King of the Hill*, see <https://youtu.be/VVAfeObAZzI>

posing (blue) team will always be unfamiliar though. Here the background bots were trained without any special pseudorewards.

SC 1 Focal team versus shaped bot team. Same as *SC 0* except here the background bots were trained with a special pseudoreward scheme which made them learn a spawn-camping strategy. They try to get into the opposing (red) team's base and zap players there as soon as they spawn.

SC 2 Ad hoc teamwork with default bots. In this scenario all players are sampled from the background population except for one focal agent, who joins the red team. The focal player must coordinate with its teammates to help them win the game against the opposing (blue) team, despite having never played with them or any of the opponent players before. Here the background bots were trained without any special pseudorewards.

SC 3 Ad hoc teamwork with shaped bots. Same as *SC 2* except here the background bots were trained with a special pseudoreward scheme which made them learn a spawn-camping strategy.

Predator-Prey *

There are two roles: predators and prey. The prey try to eat apples and acorns. The predators try to eat prey.

Apples are worth 1 reward and can be eaten immediately. Acorns are worth 18 reward but they take a long time to eat. It is not possible to move while eating so a prey player is especially vulnerable while they eat it.

Predators can also eat other predators, though they get no reward for doing so. However, a predator might eat another predator anyway in order to remove a competitor who might otherwise eat its prey.

When prey travel together in tight groups they can defend themselves from being eaten by predators. When a predator tries to eat its prey then all other prey who are not currently eating an

acorn within a radius of 3 are counted. If there are more prey than predators within the radius then the predator cannot eat the prey.

So prey are safer in groups. However, they are also tempted to depart from their group and strike out alone since that way they are more likely to be the one to come first to any food they find.

Both predators and prey have limited stamina. They can only move at top speed for a limited number of consecutive steps, after which they must slow down. Stamina is visible to all with a colored bar above each player's head. If the bar over a particular player's head is invisible or green then they can move at top speed. If it is red then they have depleted their stamina and can only move slowly until they let it recharge. Stamina is recharged by standing still for some number of consecutive time steps, how many depends on how much stamina was depleted. Predators have a faster top speed than prey but they tire more quickly.

Prey but not predators can cross tall grass (green grassy locations). Prey must still be careful on grass though since predators can still reach one cell over the border to eat prey on the edge of safety.

Both predators and prey respawn 200 steps after being eaten.

Predator Prey: Alley Hunt²⁸

In this variant prey must forage for apples in a maze with many dangerous dead-end corridors where they could easily be trapped by predators.

Scenarios

SC 0 Focal prey visited by background predators.

This is a resident mode scenario where eight focal prey are joined by five background predators. The focal prey must work together to prevent predation long enough to escape the chokepoint near the safety grass.

SC 1 Focal predators aim to eat resident prey. This is a visitor mode scenario where five focal

²⁸For a video of Predator Prey: Alley Hunt, see <https://youtu.be/ctVjhN7VYgo>

predators hunt eight background prey. The focal predators must coordinate to prevent their prey from escaping the chokepoint near the safety grass where they spawn. It is especially important for predators to resist the temptation to eat one another since having less predators near the chokepoint makes it easier for prey to escape.

SC 2 *A focal predator competes with background predators to eat prey.* This is a visitor mode scenario where one focal predator joins twelve background players, four predators and eight prey. Here the focal predator must ad hoc collaborate with unfamiliar background predators to hunt unfamiliar background prey.

SC 3 *One focal prey ad hoc cooperates with background prey to avoid predation.* This is visitor mode scenario where one focal prey joins twelve background players, five predators and seven prey. Here the focal prey must ad hoc collaborate with unfamiliar background prey to evade predation from unfamiliar background predators.

Predator Prey: Open²⁹

In this variant prey must forage over a large field of apples and acorns in the center of the map. Since the space is so open it should be possible for the prey to move together in larger groups so they can defend themselves from predators. Another prey strategy focused on acorns instead of apples is also possible. In this case prey collect acorns and bring them back to safe tall grass to consume them.

Scenarios

SC 0 *Focal prey visited by background predators.* This is a resident mode scenario where ten focal prey are joined by three background predators. The focal prey could evade predation by moving together as a “herd”, grazing on apples while always staying close enough to one another to prevent predation.

²⁹For a video of Predator Prey: Open, see <https://youtu.be/OZlrkWsWzMw>

Alternatively, some focal players may focus on collecting acorns and bringing them to safety for consumption.

SC 1 *Focal predators aim to eat basic resident prey.* This is a visitor mode scenario where three focal predators hunt ten background prey.

SC 2 *A focal predator competes with background predators to eat prey.* This is a visitor mode scenario where one focal predator joins twelve background players, two other predators and ten prey.

SC 3 *One focal prey ad hoc cooperates with background prey to avoid predation.* This is visitor mode scenario where one focal prey joins twelve background players, three predators and nine other prey. It may require ad hoc coordinated movement with the other prey to remain in a large enough group to evade predation. However the acorn-focused strategy is relatively individualistic so it can be accomplished without too much coordination.

SC 4 *Focal predators hunt smarter resident prey.* This is a visitor mode scenario where three focal predators hunt ten background prey. The prey all use the strategy of collecting acorns and bringing them back to safety for consumption.

SC 5 *A focal predator competes with background predators to hunt smarter prey.* This is a visitor mode scenario where one focal predator joins twelve background players, two other predators and ten prey. The prey all use the strategy that focuses on collecting acorns and bringing them back to safety for consumption.

SC 6 *One focal prey ad hoc cooperates with background smart prey to avoid predation.* This is visitor mode scenario where one focal prey joins twelve background players, three predators and nine other prey. The background prey all focus on collecting acorns and bringing them back to safety for consumption.

Predator Prey: Orchard³⁰

In this variant there are two areas of the map containing food: an apple-rich region to the north of the safe tall grass and an acorn-rich region to the east. There are two possible prey strategies focusing on either apples or acorns. However, in this case it is clear that focusing on acorns is the better strategy since they are relatively close to the safe tall grass. They can easily be collected and brought back to safety for consumption.

Scenarios

SC 0 *Focal prey visited by background predators.*

This is a resident mode scenario where eight focal prey are joined by five background predators. The focal prey are best served by a strategy focused on collecting acorns and bringing them back to safety for consumption.

SC 1 *Focal predators aim to eat resident population of unspecialized prey.*

This is a visitor mode scenario where five focal predators hunt eight background prey. The focal predators must coordinate to guard both directions that the prey may escape the safe grassy area where most of them spawn. Most predators should stay in the chokepoint leading to the apple-rich region since most prey will try a strategy that requires them to pass that way. At least one predator must guard the acorn-rich region since some prey in this background population do use an acorn-focused strategy.

SC 2 *A focal predator competes with background predators to eat unspecialized prey.*

This is a visitor mode scenario where one focal predator joins twelve background players, four predators and eight prey. Here the focal predator must go to either the apple-rich region's chokepoint or the acorn-rich region as needed, depending on what the other predators do.

SC 3 *One focal prey ad hoc cooperates with unspecialized background prey to avoid predation.*

This is visitor mode scenario where one focal prey joins twelve background players, five predators and seven prey. Here the focal prey must ad hoc collaborate with unfamiliar background prey, this may require dynamically switching between apple-focused and acorn-focused strategies, as needed, depending on where there are more predators and prey.

SC 4 *Focal predators aim to eat resident population of acorn specialist prey.* This is a visitor mode scenario where five focal predators hunt eight background prey. The focal predators must coordinate to guard both directions that the prey may escape the safe grassy area where most of them spawn. Since some prey occasionally spawn in the apple-rich region in the upper part of the map, some predators must be ready to go after them there. Most predators should stay near the acorns since most prey specialize in an acorn-focused strategy.

SC 5 *A focal predator competes with background predators to eat acorn specialist prey.* This is a visitor mode scenario where one focal predator joins twelve background players, four predators and eight prey. Here the focal predator must go to either the apple-rich region's chokepoint or the acorn-rich region as needed, depending on what the other predators do. In this case, more predators should stay near the acorn-rich region since more prey are focused that way here.

SC 6 *One focal prey ad hoc cooperates with acorn specialized background prey to avoid predation.* This is visitor mode scenario where one focal prey joins twelve background players, five predators and seven prey. In this case the best strategy is probably to collect acorns and bring them to safety for consumption. This strategy likely works best when all agents apply it since if they do then they are often near enough to one another to lend support and prevent predation.

³⁰For a video of Predator Prey: Orchard, see <https://youtu.be/gtd-ziZYJRI>

Predator Prey: Random Forest³¹

In this variant there are only acorns, no apples. And, there is no fully safe tall grass. The tall grass that there is on this map is never large enough for prey to be fully safe from predation. The grass merely provides an obstacle that predators must navigate around while chasing prey.

Scenarios

SC 0 *Focal prey visited by background predators.*

This is a resident mode scenario where eight focal prey are joined by five background predators. The focal prey must run through randomly located small patches of tall grass—which slow down predators since they must go around them—and stay close to other prey in order to evade predators long enough to collect acorns.

SC 1 *Focal predators aim to eat resident prey.* This is a visitor mode scenario where five focal predators hunt eight background prey. It is especially important in this scenario for the focal predators to manage their stamina properly.

SC 2 *A focal predator competes with background predators to eat prey.* This is a visitor mode scenario where one focal predator joins twelve background players, four predators and eight prey.

SC 3 *One focal prey ad hoc cooperates with background prey to avoid predation.* This is visitor mode scenario where one focal prey joins twelve background players, five predators and seven prey.

Territory *

Players cannot walk through resources, they are like walls.

Players can claim a resource in two ways: (1) by touching it, and (2) by using a “claiming beam”, different from the zapping beam, which they also have. Claimed resources are colored in

the unique color of the player that claimed them. Unclaimed resources are gray.

Once a resource has been claimed a countdown begins. After 100 timesteps, the claimed resource becomes active. This is visualized by a white and gray plus sign appearing on top. Active resources provide reward stochastically to the player that claimed them at a rate of 0.01 per timestep. Thus the more resources a player claims and can hold until they become active, the more reward they obtain.

The claiming beam is of length 2. It can color through a resource to simultaneously color a second resource on the other side. If two players stand on opposite sides of a wall of resources of width 2 and one player claims all the way across to the other side (closer to the other player than themselves) then the player on the other side might reasonably perceive that as a somewhat aggressive action. Less aggressive of course than the other option both players have: using their zapping beam. If any resource is zapped twice then it gets permanently destroyed. It no longer functions as a wall or a resource, allowing players to pass through.

Like resources, when players are hit by a zapping beam they also get removed from the game and never regenerate. This is different from other substrates where being hit by a zapping beam is not permanent removal. In territory once a player has been zapped out it is gone. All resources it claimed are immediately returned to the unclaimed state.

Territory: Inside Out³²

Five individuals can claim resource walls for reward by painting them in their own color. The players all spawn on the outside of a randomly generated maze of resource walls which differs from episode to episode.

Scenarios

SC 0 *Focals are resident and visited by an aggressor.* This is a resident mode scenario where

³¹For a video of Predator Prey: Random Forest, see https://youtu.be/ZYkXwvn5_Sc

³²For a video of Territory: Inside Out, see <https://youtu.be/LdbIjnHaisU>

four focal players are visited by one very aggressive background player. The background bot was trained using the ACB algorithm and did not use any special pseudorewards. It became aggressive because it was co-trained alongside others who had an extra pseudoreward for zapping. This training scheme for the background population was selected so the bots would learn to fight aggressively but still maintain focus on the substrate's true reward structure. The result was a background bot who frequently invades the territory of its neighbors and claims their resources. In this scenario the focal population, who are resident, should defend themselves against invasion, or potentially flee to unoccupied areas elsewhere on the map if attacked.

SC 1 *Visiting a population of aggressors.* This is a visitor mode scenario where one focal player visits four aggressive background players. The background population was trained with the method described in SC 0. In this scenario the lone focal player must defend its territory from potentially any direction, or be willing and able to flee to other unclaimed areas if they are available.

SC 2 *Focals are resident, visited by a bot that does nothing.* This is a resident mode scenario where four focal players are visited by a bot that does nothing. Since the background player does not do anything, it can easily be zapped out by any other player. This provides focal players with an opportunity to capture more territory since they need only split it among four instead of five. This windfall may destabilize tacit cooperation, especially if some players end up taking more than others, or if borders become displaced from their locations in training. The scenario is testing whether the focal population can cope with such a disturbance.

SC 3 *Focals visit a resident population that does nothing.* In this visitor mode scenario, all four of the background players do nothing. It is best to understand this scenario in the context of SC 2. If a focal population does well in SC 2 then it might not have noticed

that it was facing a co-player who did not use any resources or defend itself. This scenario shows that when faced with a very quiet map, where no bots move or claim any resources, the focal agent can take advantage of the benign situation and get a very high reward. If a focal population performs well in SC 2 but does not perform well in this scenario then it's likely that it simply did not respond to the opportunity provided by the presence of the null co-player in SC 2.

SC 4 *Focals are resident, visited by a bot that claims a moderate size territory and mostly tolerates its neighbors.* This is a resident mode scenario where four focal players are visited by an ACB bot who does not claim too much territory and is not very aggressive. Tolerance of neighbors is a good strategy here.

SC 5 *Focals visit a resident population that claims a moderate size territory and mostly tolerates its neighbors.* In this visitor mode scenario one focal player meets four background players who do not claim too much territory and are not very aggressive. The background players will sometimes defend themselves if attacked, but they mostly tolerate one another and do not claim very large territories. The focal agent can do well by claiming a large territory for itself.

Territory: Open³³

Nine individuals can claim resources for reward by painting them in their own color. All players spawn in an open space near one another and at some distance away from all the resources. Some parts of the map are more resource rich than others.

Scenarios

SC 0 *Focals are resident and visited by an aggressor.* This is a resident mode scenario where eight focal players are visited by one very

³³For a video of Territory: Open, see <https://youtu.be/F1006LFIZHI>

aggressive background player. The background bot was trained using the ACB algorithm and did not use any special pseudorewards. It became aggressive because it was co-trained alongside others who had an extra pseudoreward for zapping. This training scheme for the background population was selected so the bots would learn to fight aggressively but still maintain focus on the substrate's true reward structure. The result was a background bot who frequently attacks other players and tries to claim their resources. In this scenario the focal population, who are resident, should be prepared to defend themselves or give up their resources and flee when it is safe to do so and other resources are unclaimed elsewhere.

SC 1 *Visiting a population of aggressors.* This is a visitor mode scenario where one focal player visits eight aggressive background players. The background population was trained with the method described in SC 0 so they are very aggressive. In this scenario the lone focal player must defend itself and its resources from all other players it encounters.

SC 2 *Focals are resident, visited by a bot that does nothing.* This is a resident mode scenario where eight focal players are visited by a bot that does nothing. Since the background player does not do anything, it can easily be zapped out by any other player. This may ultimately provide focal players with an opportunity to capture more territory since they need only split it among eight instead of nine. It should be easy to get a good score in this scenario since the effect of the null bot will be minimal on this open map.

SC 3 *Focals visit a resident population that does nothing.* In this visitor mode scenario, all eight of the background players do nothing. This scenario tests that the focal agent really knows how rewards are obtained in this substrate. At the start of the episode they will see that the other players are not moving and vulnerable to being zapped out.

They will also see that all the resources remain unclaimed. If they really understand the game dynamics then they will see that it is rational in this situation to claim a very large share of the map's resources, even all the resources if they already zapped out all their potential competitors at the start. The focal agent should be able to take advantage of this very benign situation to get a very high reward.

Territory: Rooms³⁴

Same dynamics as *Territory Open* except that individuals start in segregated rooms that strongly suggest a partition individuals could adhere to. They can break down the walls of these regions and invade each other's "natural territory", but the destroyed resources are lost forever. A peaceful partition is possible at the start of the episode, and the policy to achieve it is easy to implement. But if any agent gets too greedy and invades, it buys itself a chance of large rewards, but also chances inflicting significant chaos and deadweight loss on everyone if its actions spark wider conflict. The reason it can spiral out of control is that once an agent's neighbor has left their natural territory then it becomes rational to invade the space, leaving one's own territory undefended, creating more opportunity for mischief by others.

Scenarios

SC 0 *Focals are resident and visited by an aggressor.* This is a resident mode scenario where eight focal players are visited by one very aggressive background player. The background bot was trained using the ACB algorithm and did not use any special pseudorewards. It became aggressive because it was co-trained alongside others who had an extra pseudoreward for zapping. This training scheme for the background population was selected so the bots would learn to fight aggressively but still maintain focus on the substrate's true reward structure. The result was a background bot who frequently

³⁴For a video of *Territory: Rooms*, see <https://youtu.be/4URkGR9iv9k>

invades the territory of its neighbors and claims their resources. In this scenario the focal population, who are resident, should defend themselves against invasion.

- SC 1 *Visiting a population of aggressors.* This is a visitor mode scenario where one focal player visits eight aggressive background players. The background population was trained with the method described in SC 0. In this scenario the lone focal player must defend its territory from potentially any direction.
- SC 2 *Focals are resident, visited by a bot that does nothing.* This is a resident mode scenario where eight focal players are visited by a bot that does nothing. Since the background player does not do anything, it can easily be zapped out by any other player. This provides focal players with an opportunity to capture more territory since they need only split it among eight instead of nine. This windfall may destabilize tacit cooperation, since whichever player takes over the background bot's territory is likely to become overextended with too large a territory to easily defend. This situation could tempt other neighbors to invade, causing a cascade of conflict. The scenario is testing whether the focal population can cope with such a disturbance.
- SC 3 *Focals visit a resident population that does nothing.* In this visitor mode scenario, all eight of the background players do nothing. It is best to understand this scenario in the context of SC 2. If a focal population does well in SC 2 then it might not have noticed that it was facing a co-player who did not use any resources or defend itself. This scenario shows that when faced with a very quiet map, where no bots move or claim any resources, the focal agent can take advantage of the benign situation and get a very high reward. If a focal population performs well in SC 2 but does not perform well in this scenario then it's likely that it simply did not respond to the opportunity provided by the presence of the null co-player in SC 2.

* in the Matrix

This mechanism was first described in [Vezhnevets et al. \(2020\)](#).

Agents can move around the map and collect resources of K discrete types. In addition to movement, the agents have an action to fire their “interaction beam”. All agents carry an inventory with the count of resources picked up since last respawn. The inventory is represented by a vector

$$\rho = (\rho_1, \dots, \rho_K).$$

Agents can observe their own inventory but not the inventories of their coplayers. An interaction occurs whenever any player zaps any other player with their interaction beam. The resolution of the interaction is driven by a traditional matrix game, where there is a payoff matrix A describing the reward produced by the pure strategies available to the two players. Each player gets rewarded corresponding to the expected payoff they would have gotten by playing a mixed strategy in the underlying matrix game determined by the objects in their inventory.

Resources map one-to-one to pure strategies of the matrix game. For the purpose of resolving the interaction, the zapping agent is considered the row player, and the zapped agent the column player when the underlying matrix game is symmetric. A different rule for assigning which player is the row versus column player is used in the case of an asymmetric underlying game (see *Bach or Stravinsky in the matrix* below). The mixed strategy a player executes in the underlying matrix game depends on the resources they picked up before the interaction. The more resources of a given type a player picks up, the more committed they becomes to the pure strategy corresponding to that resource. In particular, an agent with inventory ρ plays the mixed strategy with weights

$$v = (v_1, \dots, v_K)$$

where

$$v_i = \frac{\rho_i}{\sum_{j=1}^K \rho_j}.$$

The rewards r_{row} and r_{col} for the (zapping) row and the (zapped) column player, respectively, are

assigned via

$$\begin{aligned} r_{\text{row}} &= \nu_{\text{row}}^T A_{\text{row}} \nu_{\text{col}} \\ r_{\text{col}} &= \nu_{\text{row}}^T A_{\text{col}} \nu_{\text{col}} \end{aligned}$$

If the game is symmetric then $A_{\text{row}} = A_{\text{col}}^T$. These reward calculation definitions mirror those used in evolutionary game theory (Weibull, 1997).

To obtain high rewards, an agent could either collect resources to ensure playing a Nash strategy in the matrix game, or correctly identify what resource its interaction partner is collecting and collect the resources that constitute a best response. Some scenarios have two players while others have eight players. Whenever there are more than two players then there is an element of partner choice. Interactions in these substrates are always dyadic but players can decide who from the group to interact with by zapping them with their interaction beam.

After interacting both players get removed from the game for some number of steps: 50 steps for Arena substrates and 5 steps for Repeated substrates. Except for the case of *Running with Scissors in the matrix: One Shot*, where episodes end after the first interaction, in all other substrates the episodes last for at least 1000 timesteps. After 1000 steps they have a constant probability of 0.1 of ending after every subsequent 100 step interval. Thus for the Repeated substrates there will usually be more than 50 interactions per episode once agents are trained. Less trained agents typically take more time to interact and thus experience fewer interactions per episode. After each interaction the inventory of both players is reset to its initial value: a vector of all 1s.

Bach or Stravinsky in the matrix: Arena³⁵

Individuals collect resources that represent ‘Bach’ or ‘Stravinsky’ choices in the classic Bach or Stravinsky matrix game (which was called Battle of the Sexes in Luce and Raiffa (1957/1989)). As with all * in the matrix substrates, when a pair of players interact their normalized inventories

³⁵For a video of *Bach or Stravinsky in the matrix: Arena*, see <https://youtu.be/QstXaLjiqK4>

are converted to components of a mixed strategy in the repeated game. The rewards obtained by each player are congruent with those of players in the matrix game employing the mixed strategy they selected. Bach or Stravinsky exposes a tension between reward for the group and fairness between individuals.

Unlike other * in the matrix games, Bach or Stravinsky is asymmetric. Half the players (blue avatars) are assigned to always be the row player in all their interactions, and the other half (orange avatars) are assigned to always be the column player. The row players have the role ‘bach fan’ whereas the column players have the role ‘stravinsky fan’. There is no effect when two players with the same row/column assignment try to interact with one another. The game only resolves when row and column players interact with one another. The winner’s inventory is also reset after an interaction. Because this game is asymmetric, there is a different matrix A_{row} for the row player and A_{col} for the column player. The matrices for the interaction are:

$$A_{\text{row}} = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix},$$

and

$$A_{\text{col}} = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}.$$

Scenarios

SC 0 *Visiting background population who picks Bach*. This is a visitor mode scenario, where one focal player, who is a Bach fan, is visiting a sample of 7 players from the background population that all choose Bach. The background population is trained to choose Bach regardless of their preference. In this scenario the focal player must follow the consensus and choose Bach to maximise their score. This is consistent with their role preference as a Bach fan.

SC 1 *Visiting background population who picks Stravinsky*. This is a visitor mode scenario, where one focal player, who is a Bach fan, is visiting a sample of 7 players from the background population that all choose Stravinsky. The background population is trained

to choose Stravinsky regardless of their preference. In this scenario the focal player must yield to the consensus and choose Stravinsky to maximise their score. This is in conflict with their role preference as a Bach fan.

SC 2 Visited by a pure bot. This is a resident mode scenario, where a sample of 7 focal players is visited by a single background bot. The bot is sampled from a background population of bots that are trained to always pick the same recourse (Bach or Stravinsky). The bot is sampled into a player slot that prefers Stravinsky. In this scenario, the focal population should stick with their chosen equilibrium (if they have one). This tests for stability of the equilibrium learnt by the focal population.

SC 3 Visited by three pure Bach pickers. This is a resident mode scenario, where a sample of 7 focal players is visited by three background bots. All the background bots are trained to prefer Bach, although they are sampled into player slots with Stravinsky preference. In this scenario, the focal population should stick with their chosen equilibrium (if they have one). This strongly tests for stability of the equilibrium learnt by the focal population.

SC 4 Visited by three pure Stravinsky pickers. This is a resident mode scenario, where a sample of 7 focal players is visited by three background bots. All the background bots are trained to prefer Stravinsky and they are sampled into player slots with Stravinsky preference. In this scenario, the focal population should stick with their chosen equilibrium (if they have one). This strongly tests for stability of the equilibrium learnt by the focal population.

SC 5 Visiting background population who alternate, starting from Stravinsky, repeating each twice. This is a visitor mode scenario, where one focal player, who is a Bach fan, is visiting a sample of 7 players from the background population that have alternating preference. The background bots

are trained to alternate between Bach and Stravinsky, starting from Stravinsky. After picking and playing Stravinsky twice, they are trained to switch to Bach for two iterations and so on. In this scenario, the focal population should follow the moving convention, identifying which resource is favoured at the moment.

SC 6 Visiting background population who alternate, starting from Bach, repeating each twice. This is a visitor mode scenario, where one focal player, who is a Bach fan, is visiting a sample of 7 players from the background population that have alternating preference. The background bots are trained to alternate between Bach and Stravinsky, starting from Bach. After picking and playing Stravinsky twice, they are trained to switch to Bach for two iterations and so on. In this scenario, the focal population should follow the moving convention, identifying which resource is favoured at the moment.

Bach or Stravinsky in the matrix: Repeated³⁶

See the description of *Bach or Stravinsky in the matrix: Arena*. The difference between the two variants is that *Repeated* is a two-player game while *Arena* is an eight-player game.

Scenarios

SC 0 Meeting a stubborn Bach picker. In this scenario the focal player meets a background bot that is trained to always pick Bach. The bot is a Bach fan while the focal player is a Stravinsky fan. The focal player has to identify the stubbornness (Bach).

SC 1 Meeting a bot who plays Bach despite not being a fan. In this scenario the focal player meets a background bot that is trained to always pick Bach. The bot is a Stravinsky fan while the focal player is a Bach fan. The focal player can simply play to their preference (Bach).

³⁶For a video of *Bach or Stravinsky in the matrix: Repeated*, see <https://youtu.be/I2uYugpdfFQ>

SC 2 *Meeting a bot who plays Stravinsky despite not being a fan.* In this scenario the focal player meets a background bot that is trained to always pick Stravinsky. The bot is a Bach fan while the focal player is a Stravinsky fan. The focal player can simply play to their preference (Stravinsky).

SC 3 *Meeting a stubborn Stravinsky picker.* In this scenario the focal player meets a background bot that is trained to always pick Stravinsky. The bot is a Stravinsky fan while the focal player is a Bach fan. The focal player has to identify the stubbornness of the bot and yield to the bot's convention (Stravinsky).

SC 4 *Bach fan focal agent meets an imperfectly copying partner.* In this scenario a focal players meets a background bot that is trained to repeat the last choice of the focal agent. The bot is a Stravinsky fan while the focal player is a Bach fan. The focal player can stick to their preference (Bach).

SC 5 *Stravinsky fan focal agent meets an imperfectly copying partner.* In this scenario a focal players meets a background bot that is trained to repeat the last choice of the focal agent. The bot is a Bach fan while the focal player is a Stravinsky fan. The focal player can stick to their preference (Stravinsky).

SC 6 *Bach fan focal agent meets a turn-taking partner.* In this scenario the focal player meets a background bot that is trained to alternate between resources after each interaction. The bot is a Stravinsky fan while the focal player is a Bach fan. The focal player should alternate with the bot.

SC 7 *Bach fan focal agent meets a turn-taking partner who repeats each goal/resource three times before switching.* In this scenario the focal player meets a background bot that is trained to alternate between resources, repeating the choice for three interactions before alternating . The bot is a Stravinsky fan while the focal player is a Bach fan. The focal player should alternate with the bot.

Chicken in the matrix: Arena³⁷

Individuals can gather resources of different colors. Players' encounters are resolved with the same payout matrix as the game 'Chicken', in which both players attempting to take advantage of the other leads to the worst outcome for both. Collecting red resources pushes one's strategy choice toward playing 'hawk' while collecting green resources pushes it toward playing 'dove'. The matrix for the interaction is:

$$A_{\text{row}} = A_{\text{col}}^T = \begin{bmatrix} 3 & 2 \\ 5 & 0 \end{bmatrix} .$$

Scenarios

SC 0 *Visiting unconditional dove players.* This is a visitor scenario, where a single focal player is visiting 7 background bots trained to always play dove. The focal player can score high by playing hawk.

SC 1 *Focals are resident and visitors are unconditional dove players.* This is a resident mode scenario, where 5 focal players are visited by 3 background bots trained to always play dove. The focal players can score high by playing hawk.

SC 2 *Focals are resident and visitors are unconditional hawk players.* This is a resident mode scenario, where 5 focal players are visited by 3 background bots trained to always play hawk. The focal players has to play dove.

SC 3 *Visiting a population of hair-trigger grim reciprocator bots who initially cooperate but, if defected on once, will retaliate by defecting in all future interactions.* This is a visitor scenario, where a single focal player is visiting 7 background bots trained to perform the "one-strike grim trigger" strategy—the bots will initially play dove, but after encountering a hawk once they switch to hawk for the rest of the episode. To maximise their score, the focal agent has to play dove.

SC 4 *Visiting a population of two-strikes grim reciprocator bots who initially cooperate but, if*

³⁷For a video of *Chicken in the matrix: Arena*, see <https://youtu.be/94DHJ6BVEJM>

defected on twice, will retaliate by defecting in all future interactions. This is a visitor scenario, where a single focal player is visiting 7 background bots trained to perform the "two-strikes grim trigger" strategy—the bots will initially play dove, but after encountering a hawk *twice* they switch to hawk for the rest of the episode. To maximise their score, the focal agent can play hawk on each bot only ones, but then has to stick to dove.

SC 5 *Visiting a mixed population of k-strokes grim reciprocator bots with k values from 1 to 3, they initially cooperate but, if defected on k times, they retaliate in all future interactions.* This is a visitor scenario, where a single focal player is visiting 7 background bots trained to perform the "*k*-strikes grim trigger" strategy—the bots will initially play dove, but after encountering a hawk *k* times they switch to hawk for the rest of the episode. The latent variable *k* is sampled from {1, 2, 3}. To maximise their score, the focal agent can play hawk on each bot *k* times, but then has to stick to dove.

SC 6 *Visiting a mixture of pure hawk and pure dove players.* This is a visitor scenario, where 3 focal agents are visiting a background population of 5 bots. The background population consists of bots trained to exclusively pursue dove or hawk recourse. The bots are sampled at random. For the focal players to maximise their score, they have to identify the dove and hawk players and play hawk / dove against them respectively.

Chicken in the matrix: Repeated³⁸

See the description of *Chicken in the matrix: Arena*. The difference between the two variants is that *Repeated* is a two-player game while *Arena* is an eight-player game.

Scenarios

SC 0 *Partner may play either hawk or dove.* Here

³⁸For a video of *Chicken in the matrix: Repeated*, see <https://youtu.be/bFwV-udmRb4>

the focal agent plays with a partner sampled from the background population that was trained to play a pure strategy: either to collect mainly hawk or mainly dove resources. However, the specific background player is sampled at test time so it could of either type. All background bots commit strongly to their choice, aiming to collect at least five (or seven) resources before interacting. To get a high score the focal agent should scout out which pure strategy its partner is playing and then pick the other pure strategy. So if the background bot plays dove the focal agent should play hawk. If the background bot plays hawk then the focal agent should play dove.

SC 1 *Partner typically plays dove.* Here the focal player meets a partner sampled from the background population that was trained to always play dove and commits to it strongly, aiming to collect either five or seven resources before interacting. The focal player can get a high score by playing hawk.

SC 2 *Partner typically plays hawk.* Like **SC 1** but here the background bot plays hawk so the focal agent should play dove.

SC 3 *Partner is a hair-trigger grim reciprocator, i.e. one who initially cooperates but, if defected on once, will retaliate by defecting forever after.* The focal agent is successful when it mostly plays dove.

SC 4 *Partner is a two-strokes grim reciprocator, i.e. one who initially cooperates, but if defected on twice, will retaliate by defecting forever after.* The focal agent is successful when it mostly plays dove.

SC 5 *Partner is a tit-for-tat conditional cooperator.* The focal agent is successful when it alternates between hawk and dove along with its partner.

SC 6 *Partner is a tit-for-tat conditional cooperator who occasionally plays hawk instead of dove.* Same as **SC 5** but in this case must avoid being distracted by background bot's more noisy strategy and get quickly reestablish the alternation rhythm after each.

SC 7 *Partner plays dove for a while then switches to hawk.* A good score would be obtained by a focal agent that initially plays hawk but then switches to dove at the time its partner switches to hawk.

SC 8 *Partner tries to take advantage of the focal player by playing hawk, but if punished, partner then switches to tit-for-tat conditional cooperation.* A good focal agent would punish its partner to get it into the tit-for-tat regime and then alternate playing hawk and dove.

SC 9 *Partner tries to take advantage of the focal player by playing hawk, but if punished, partner then switches to noisy tit-for-tat conditional cooperation.* Same as SC 8 but in this case must avoid being distracted by background bot's more noisy strategy and get quickly reestablish the alternation rhythm after each.

Prisoners Dilemma in the matrix: Arena³⁹

Eight individuals collect resources that represent ‘defect’ (red) or ‘cooperate’ (green) and compare inventories in an encounter. Consequences of the inventory comparison are congruent with the classic *Prisoner’s Dilemma* matrix game. This game exposes tension between reward for the group and reward for the individual. The matrix for the interaction is

$$A_{\text{row}} = A_{\text{col}}^T = \begin{bmatrix} 3 & 0 \\ 5 & 1 \end{bmatrix} .$$

Scenarios

SC 0 *Visiting unconditional cooperators.* This is a visitor mode scenario. A single focal player visits a population of background players who strongly prefer to collect cooperation resources and then interact. The highest possible reward is obtained by unconditionally defecting (collecting as many defect resources as possible and then interacting). However, this presupposes that the focal knows the background players are unconditional cooperators, which they would not

³⁹For a video of *Prisoners Dilemma in the matrix: Arena*, see <https://youtu.be/81QrMpsP-HU>

know under test conditions. Since quite a few players are collecting cooperate resources simultaneously they often become depleted. This makes it relatively obvious to the focal player that others are not defecting. Unfortunately, for cooperators, the scarcity of cooperate resources in this situation makes it difficult to get a high score by cooperating.

SC 1 *Focals are resident and visited by an unconditional cooperator.* This is a resident mode scenario. Seven focal players are joined by a single background player who strongly prefers to collect cooperation resources. Most of the focal population’s collective return will be determined by how well focal players cooperate with one another.

SC 2 *Focals are resident and visitors defect unconditionally.* This is a resident mode scenario. Six focal players are joined by two background players who strongly prefer to collect defection resources. The focal players should cooperate with other focals, and defect towards background payers. Because the focals don’t know *a priori* the strategy of the background players, they probably should cooperate first, and only switch to defection against those who defected against them.

SC 3 *Visiting a population of hair-trigger grim reciprocator bots who initially cooperate but, if defected on once, will retaliate by defecting in all future interactions.* This is a visitor mode scenario. A single focal player visits seven background players who initially cooperate (preferentially collecting cooperation resources), but if at any point a co-player defects to them, they will switch to defecting for the rest of the episode. A focal population that cooperates (perhaps conditionally) should achieve a high score.

SC 4 *Visiting a population of two-strikes grim reciprocator bots who initially cooperate but, if defected on twice, will retaliate by defecting in all future interactions.* This is a visitor mode scenario. Like the previous sce-

nario, but the Grim reciprocators in the background population are more generous, allowing for more forgiving exploration of their exploitability.

SC 5 *Visiting a mixed population of k-strokes grim reciprocator bots with k values from 1 to 3, they initially cooperate but, if defected on k times, they retaliate in all future interactions.* This is a visitor mode scenario. In this case, three focal players visit five background players. Focal players should cooperate with one another. As in the previous two scenarios, a focal population of conditional cooperators should achieve a high score.

Prisoners Dilemma in the matrix: Repeated⁴⁰

See the description of *Prisoners Dilemma in the matrix: Arena*. The difference between the two variants is that *Repeated* is a two-player game while *Arena* is an eight-player game.

Scenarios

SC 0 *Partner may play either cooperate or defect.* The optimal strategy is simply to unconditionally defect. However, given that the focal doesn't know the strategy of the background player, a good strategy is more subtle. A reasonable strategy is to be a grim reciprocator cooperator, which would cooperate with the cooperator, and defect to the defector. Alternatively the focal player might try to ascertain whether the background player is exploitable. Doing so, however, carries a risk, for if the background player were to be a Grim reciprocator (like in other scenarios), this would cause them to defect for the rest of the episode.

SC 1 *Partner typically plays cooperate.* The optimal strategy is simply to unconditionally defect. The same considerations about uncertainty of the background player's strategy from Scenario 0 apply here.

SC 2 *Partner typically plays defect.* The optimal strategy is simply to unconditionally defect. However, because the focal player doesn't *a priori* know the strategy of the background player, they must first try to find out their strategy. This can be done by looking at which resources they collect or by paying attention to the results of the first few interactions. Once the focal has identified its background partner is defecting then it may have confidence that it should defect as well. The focal player should also consider the possibility that the background bot is corrigible, i.e. that it could be persuaded to switch from defection to cooperation. This is not the case here but the background populations used in SC 8 and SC are corrigible.

SC 3 *Partner is a hair-trigger grim reciprocator, i.e. one who initially cooperates but, if defected on once, will retaliate by defecting forever after.* The optimal strategy is simply to cooperate. Grim reciprocator background players are non-exploitable, and there is no way to know how they will react to a defection ahead of time. Because of this uncertainty, testing for exploitability can lead to poor performance of the focal player. Conditional cooperators who cooperate first but retaliate if defected on should achieve a high score.

SC 4 *Partner is a two-strokes grim reciprocator, i.e. one who initially cooperates, but if defected on twice, will retaliate by defecting forever after.* The optimal strategy is simply to cooperate. Grim reciprocator background players are non-exploitable, and there is no way to know how they will react to a defection ahead of time. Because of this uncertainty, testing for exploitability can lead to poor performance of the focal player. In principle, it would be possible to defect once against the background player leading to higher reward. But since it is not possible to know the background player is a two-strokes grim reciprocator, and testing it against a hair-trigger grim reciprocator leads to defection, in practice is better simply to cooperate. Conditional cooperators who cooperate first

⁴⁰For a video of *Prisoners Dilemma in the matrix: Repeated*, see <https://youtu.be/AAd9UcP0nk0>

but retaliate if defected on should achieve a high score.

SC 5 *Partner is a tit-for-tat conditional cooperator.* The optimal strategy is simply to cooperate. Defecting against a tit-for-tat agent, even occasionally, might lead to miscoordinated interactions where one player cooperates and the other defects, in an alternating way. Forgiveness is one way to break out of such cycles of recrimination. Conditional cooperators who cooperate first but retaliate if defected on should also be forgiving to ensure they do well in this scenario.

SC 6 *Partner is a tit-for-tat conditional cooperator who occasionally plays defect instead of cooperate.* Like the previous scenario, except the tit-for-tat background player occasionally will defect instead of cooperate. This is known as trembling hand in game theory. A strict tit-for-tat focal player would occasionally fall into miscoordinated interactions with the background player resulting in alternating cooperation and defection. As in SC5, focal conditional cooperators must also be forgiving to ensure they do well in this scenario. Forgiveness is even more important here since the background player will defect relatively frequently itself but will still implement tit-for-tat retaliation when defected on itself.

SC 7 *Partner plays cooperate for a while then switches to defect.* Similar considerations to the previous scenarios. A good strategy is a grim reciprocator, or tit-for-tat for the focal player. Unconditional cooperation would be exploited by the background player.

SC 8 *Partner tries to take advantage of the focal player by playing defect, but if punished, partner then switches to tit-for-tat conditional cooperation.* Related to Scenario 2, the optimal strategy is for the focal player to persuade the background player to stop defecting by punishing it through defecting itself. Once persuaded, the background player implements a conditional cooperation (tit-for-tat) strategy. So it is safe to start cooperating with them once you have verified that

they are themselves consistently cooperating.

SC 9 *Partner tries to take advantage of the focal player by playing defect, but if punished, partner then switches to noisy tit-for-tat conditional cooperation.* Like the previous scenario, except the focal player must implement a more generous form of conditional cooperation after persuading the background player to switch from defection.

Pure Coordination in the matrix: Arena⁴¹

Players—who in this case cannot be identified as individuals since they all look the same—can gather resources of three different colors. All eight individuals need to converge on collecting the same color resource to gain reward when they encounter each other and compare inventories. The matrix for the interaction is

$$A_{\text{row}} = A_{\text{col}}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} .$$

Scenarios

SC 0 *Focals are resident, a single visitor joins who may prefer any option; whichever option it prefers, it pursues it greedily.* Seven focal players are joined by one background player. The background player does not adapt its behavior very much to context, i.e. it single-mindedly collects one of the three resource types. As this is a resident mode scenario, the real test is of how well the focal population members coordinate among themselves, and whether their equilibrium is disturbed by the presence of the visitor or not.

SC 1 *Focals are resident, three visitors join who always select the same option as their partner in the previous interaction and do so without being too greedy.* This scenario tests whether the focal population's ability to coordinate is robust to the presence of visitors who try to join into their equilibrium

⁴¹For a video of Pure Coordination in the matrix: Arena, see https://youtu.be/LG_qvqujxPU

by always selecting the same resource as their partner picked on the previous round. Depending on the focal population's joint strategy, this may lead to scarcity issues with some resource types getting depleted leaving players unable to collect what they need to coordinate successfully.

- SC 2** *Focals are resident, three visitors join who always select the same option as their partner in the previous interaction and are greedy in doing so.* This is a more difficult version of SC1.
- SC 3** *Visiting a background population where all initially choose option a (without greed) and then switch to either b or c after some time.* One focal player joins seven background players in this visitor mode scenario. This scenario tests flexibility. The focal player needs to adapt accordingly after the background players switch their behavior.
- SC 4** *Visiting a background population where all initially choose option b (without greed) and then switch to either a or c after some time.* One focal player joins seven background players in this visitor mode scenario. This scenario tests flexibility. The focal player needs to adapt accordingly after the background players switch their behavior.
- SC 5** *Visiting a background population where all initially choose option c (without greed) and then switch to either a or b after some time.* One focal player joins seven background players in this visitor mode scenario. This scenario tests flexibility. The focal player needs to adapt accordingly after the background players switch their behavior.
- SC 6** *Visiting a background population where all typically choose option a and do not collect it too greedily.* In this visitor mode scenario one focal player joins seven background players. The test here is whether the focal player can ascertain the prevailing convention and join in.
- SC 7** *Visiting a background population where all typically choose option b and do not collect it too greedily.* In this visitor mode scenario

one focal player joins seven background players. The test here is whether the focal player can ascertain the prevailing convention and join in.

- SC 8** *Visiting a background population where all typically choose option c and do not collect it too greedily.* In this visitor mode scenario one focal player joins seven background players. The test here is whether the focal player can ascertain the prevailing convention and join in.

Pure Coordination in the matrix: Repeated⁴²

See the description of *Pure Coordination in the matrix: Arena*. The difference between the two variants is that *Repeated* is a two-player game while *Arena* is an eight-player game.

Scenarios

- SC 0** *Meeting any pure strategy player.* Here the focal agent plays with a partner sampled from the background population that was trained to play a pure strategy: either to collect mainly resource a, resource b, or resource c. However, the specific background player is sampled at test time so it could be of any of these types. All background bots commit strongly to their choice, aiming to collect at least five resources before interacting. To get a high score the focal agent should scout out which pure strategy its partner is playing and then pick the same pure strategy, collecting as many resources as it can.

- SC 1** *Meeting an agent who plays the best response to what the focal agent did in the last round.* This scenario ought to be easy since the background player tries to adapt itself to the focal player.

- SC 2** *Versus mixture of opponents who often flip to other strategies after some number of interactions.* This scenario tests flexibility since the focal player needs to adapt to players

⁴²For a video of *Pure Coordination in the matrix: Repeated*, see <https://youtu.be/biyhB378q58>

who suddenly change their strategy after some number of interactions.

SC 3 *Meeting an agent who almost always chooses resource a.* An easy scenario where the task is just to determine the choice always picked by the background player and act accordingly.

SC 4 *Meeting an agent who almost always chooses resource b.* An easy scenario where the task is just to determine the choice always picked by the background player and act accordingly.

SC 5 *Meeting an agent who almost always chooses resource c.* An easy scenario where the task is just to determine the choice always picked by the background player and act accordingly.

Rationalizable Coordination in the matrix: Arena⁴³

Same as Pure Coordination in the matrix except that differently colored resources are intrinsically of different values, suggesting an optimal color to converge on. The matrix for the interaction is

$$A_{\text{row}} = A_{\text{col}}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} .$$

Scenarios

SC 0 *Focals are resident, a single visitor joins who may prefer any option; whichever option it prefers, it pursues it greedily.* Seven focal players are joined by one background player. The background player does not adapt its behavior very much to context, i.e. it single-mindedly collects one of the three resource types. As this is a resident mode scenario, the real test is of how well the focal population members coordinate among themselves, and whether their equilibrium is disturbed by the presence of the visitor or not. The visitor may not select the best resource.

So this scenario is about testing whether the focal population can handle the introduction of an irrational individual.

SC 1 *Focals are resident, three visitors join who always select the same option as their partner in the previous interaction and do so without being too greedy.* This scenario tests whether the focal population's ability to coordinate is robust to the presence of visitors who try to join into their equilibrium by always selecting the same resource as their partner picked on the previous round. Depending on the focal population's joint strategy, this may lead to scarcity issues with some resource types getting depleted leaving players unable to collect what they need to coordinate successfully.

SC 2 *Focals are resident, three visitors join who always select the same option as their partner in the previous interaction and are greedy in doing so.* This is a more difficult version of SC1.

SC 3 *Visiting a background population where all initially choose option a (without greed) and then switch to either b or c after some time.* One focal player joins seven background players in this visitor mode scenario. This scenario tests flexibility. The focal player needs to adapt accordingly after the background players switch their behavior.

SC 4 *Visiting a background population where all initially choose option b (without greed) and then switch to either a or c after some time.* One focal player joins seven background players in this visitor mode scenario. This scenario tests flexibility. The focal player needs to adapt accordingly after the background players switch their behavior.

SC 5 *Visiting a background population where all initially choose option c (without greed) and then switch to either a or b after some time.* One focal player joins seven background players in this visitor mode scenario. This scenario tests flexibility. The focal player needs to adapt accordingly after the background players switch their behavior. In this

⁴³For a video of Rationalizable Coordination in the matrix: Arena, see <https://youtu.be/lXakuZhvrxo>

case the background players are switching away from the rational choice of c to one of the two irrational choices.

- SC 6 *Visiting a background population where all typically choose option a and do not collect it too greedily.* In this visitor mode scenario one focal player joins seven background players. The test here is whether the focal player can ascertain the prevailing convention and join in, even though the prevailing convention is deeply irrational.
- SC 7 *Visiting a background population where all typically choose option b and do not collect it too greedily.* In this visitor mode scenario one focal player joins seven background players. The test here is whether the focal player can ascertain the prevailing convention and join in, even though the prevailing convention is irrational.
- SC 8 *Visiting a background population where all typically choose option c and do not collect it too greedily.* In this visitor mode scenario one focal player joins seven background players. The test here is whether the focal player can ascertain the prevailing convention and join in, luckily the convention is the rational one, so this ought to be an easy test.

Rationalizable Coordination in the matrix: Repeated⁴⁴

See the description of *Rationalizable Coordination in the matrix: Arena*. The difference between the two variants is that *Repeated* is a two-player game while *Arena* is an eight-player game.

Scenarios

- SC 0 *Meeting any pure strategy player.* Here the focal agent plays with a partner sampled from the background population that was trained to play a pure strategy: either to collect mainly resource a, resource b, or resource c. However, the specific background player is sampled at test time so it could

⁴⁴For a video of *Rationalizable Coordination in the matrix: Repeated*, see <https://youtu.be/3brwR7DtxEI>

of any of these types. All background bots commit strongly to their choice, aiming to collect at least five resources before interacting. To get a high score the focal agent should scout out which pure strategy its partner is playing and then pick the same pure strategy, collecting as many resources as it can. This may be difficult since the background player sometimes picks irrational choices, nevertheless the focal player must adapt to them anyway.

- SC 1 *Meeting an agent who plays the best response to what the focal agent did in the last round.* This scenario ought to be easy since the background player tries to adapt itself to the focal player.
- SC 2 *Versus mixture of opponents who often flip to other strategies after some number of interactions.* This scenario tests flexibility since the focal player needs to adapt to players who suddenly change their strategy after some number of interactions. The background player may make irrational choices but the focal player still must adapt to them.
- SC 3 *Meeting an agent who almost always chooses resource a.* This scenario tests the focal player's ability to acquiesce to a deeply irrational convention forced by their partner.
- SC 4 *Meeting an agent who almost always chooses resource b.* This scenario tests the focal player's ability to acquiesce to an irrational convention forced by their partner.
- SC 5 *Meeting an agent who almost always chooses resource c.* This scenario tests the focal player's ability implement the rational convention. It should be easy.

Running With Scissors in the matrix: Arena⁴⁵

Same dynamics as *Running with Scissors in the matrix: Repeated* but with eight players. Each pairwise conflict is zero sum so the overall setting is one of competition. As with all zero sum games having more than two players, alliances

⁴⁵For a video of *Running With Scissors in the matrix: Arena*, see <https://youtu.be/6BL6J1bS2ce>

are possible in principle. But in this case the environment is not very conducive to forming them since rewards are always and only gained in pairwise conflicts and there does not appear to be any way to collude with others. The matrix for the interaction is:

$$A_{\text{row}} = A_{\text{col}}^T = \begin{bmatrix} 0 & -10 & +10 \\ +10 & 0 & -10 \\ -10 & +10 & 0 \end{bmatrix}.$$

Scenarios

SC 0 *Versus a background population containing bots implementing all three pure strategies.*

Here one focal player joins seven from the background population. The background population contains bots who implement all three pure strategies: rock, paper, and scissors. They may either commit to their strategy moderately (aiming to collect three resources before interacting) or more strongly (aiming to collect five). The task for the focal agent is to watch its opponents, see what strategy one of them is implementing, and act accordingly.

SC 1 *Versus gullible bots.* Here one focal player joins seven from the background population. The background population consists entirely of weak bots who were trained to best respond to agents playing pure strategies. They are weak opponents.

SC 2 *Versus mixture of opponents who play rock and some who flip to scissors after two interactions.* Here one focal player joins seven from the background population. The focal player should pay attention to what each prospective partner has collected since 2/3 of them play rock while 1/3 play scissors after the first two interactions. Choosing paper to best respond to rock is a bad choice if accidentally paired with an opponent playing scissors.

SC 3 *Versus mixture of opponents who play paper and some who flip to rock after two interactions.* Like SC 2 but with bots playing paper and bots switching from paper to rock.

SC 4 *Versus mixture of opponents who play scissors and some who flip to paper after two interactions.* Like SC 2 but with bots playing scissors and bots switching from scissors to paper.

SC 5 *Visiting a population of pure paper bots.* Here one focal player joins seven from the background population. All seven background bots play paper so the focal player can get a high score by playing scissors.

SC 6 *Visiting a population of pure rock bots.* Here one focal player joins seven from the background population. All seven background bots play rock so the focal player can get a high score by playing paper.

SC 7 *Visiting a population of pure scissors bots.* Here one focal player joins seven from the background population. All seven background bots play scissors so the focal player can get a high score by playing rock.

*Running With Scissors in the matrix: One Shot*⁴⁶

This environment first appeared in [Vezhnevets et al. \(2020\)](#). Two individuals gather rock, paper or scissor resources in the environment and can challenge others to a 'rock, paper scissor' game, the outcome of which depends on the resources they collected. It is possible—though not trivial—to observe the policy that one's partner is starting to implement and take countermeasures. This induces a wealth of possible feinting strategies. The observation window is 5×5 .

Scenarios

SC 0 *Versus mixed strategy opponent.* Here the focal agent must defeat an opponent that was trained to play a pure strategy: either rock, paper, or scissors. However, the specific opponent is sampled at test time so it could be any of those. All opponents commit strongly to their choice, aiming to collect at least three resources before interacting. To defeat them, the focal agent should

⁴⁶For a video of *Running With Scissors in the matrix: One Shot*, see <https://youtu.be/gtemAx4XEcQ>

scout out which pure strategy its opponent is playing and then collect the resources to implement its counter strategy. Since this is a one-shot interaction, success requires the focal agent to pay close attention to which resources are missing since they provide a clue to which strategy their opponent is implementing.

- SC 1 *Versus pure rock opponent.* Opponent always plays rock, and commits to it strongly, aiming to collect five resources before interacting. The focal player gets a high score when it picks paper and commits strongly to that choice.
- SC 2 *Versus pure paper opponent.* Same as SC 1 but opponent plays paper so focal player should play scissors.
- SC 3 *Versus pure scissors opponent.* Same as SC 1 but opponent plays scissors so focal player should play rock.

*Running With Scissors in the matrix: Repeated*⁴⁷

See the description of *Running With Scissors in the matrix: Arena*. The difference between the two variants is that *Repeated* is a two-player game while *Arena* is an eight-player game.

Scenarios

SC 0 *Versus mixed strategy opponent.* Here the focal agent must defeat an opponent that was trained to play a pure strategy: either rock, paper, or scissors. However, the specific opponent is sampled at test time so it could be any of those. All opponents commit strongly to their choice, aiming to collect at least five resources before interacting. To defeat them, the focal agent should scout out which pure strategy its opponent is playing and then collect the resources to implement its counter strategy.

SC 1 *Versus opponent who plays the best response to what the focal player did in the last round.*

⁴⁷For a video of *Running With Scissors in the matrix: Repeated*, see <https://youtu.be/rZH9nUKefcU>

Here the focal agent must defeat an opponent who may change their strategy with each interaction. The opponent will always select the best response to what the focal player selected in the previous interaction. For instance, if the focal player plays rock in one interaction then its opponent will play paper in the next interaction. On the first interaction of each episode it chooses one of the three pure strategies at random. The opponent always commits strongly to its choice, aiming to collect at least five resources before interacting. To win the focal player can trick its opponent into choosing a specific strategy and countering it. This requires changing strategy from interaction to interaction, cycling around the three options.

- SC 2 *Versus opponent who sometimes plays a pure strategy but sometimes plays the best response to what the focal player did in the last round.* Focal player must defeat an opponent sampled from the union of the background populations used in SC 0 and SC 1. The probability of sampling a pure opponent is 3/4 while the probability of sampling a best response opponent is 1/4.
- SC 3 *Versus mixture of opponents who often flip to other strategies after two interactions.* Focal player must defeat an opponent that may initially play any pure strategy and, with probability 1/3, may flip after the second interaction to the best response to the best response to its initial strategy. For example if it starts out playing rock then the best response to that would be paper, so after the second interaction it would switch to playing the best response to paper i.e. scissors. It only weakly commits to its strategy for the first two interactions. That is, it aims to collect only one resource before interacting. After two interactions, at the point when it changes strategy, it also starts committing more strongly to its choice, aiming to collect five resources before each interaction. With probability 2/3, the opponent instead plays a pure strategy throughout the entire episode. In half of the pure op-

ponent episodes the bot fully commits to its pure strategy, aiming to collect five resources before interacting, while the other half of the time it commits less strongly, aiming to collect only one resource before interacting. Note that opponents may be weakly committed to their strategy for the first two interactions regardless of whether they will ultimately flip strategy or not so it's not possible for the focal agent to observe weak commitment early on as a cue to predict whether or not their opponent will later flip strategies.

SC 4 *Versus mixture of opponents who either flip to another strategy after one interaction and keep it forever or continue to change, always best responding to what the focal player just did.* Two kinds of opponents are possible. Both change their strategy after the first interaction. With probability 3/4 the opponent will be a bot that flips to a different strategy after the first interaction and then follows it till the end of the episode. It always flips to the best response to the best response to its initial strategy (so if it initially plays rock then it will flip to scissors). With probability 1/4 the other kind of opponent is sampled. This opponent is identical to the one in **SC 1**. Both kinds of opponents always fully commit to their choice, aiming to collect at least five resources before interacting so its not possible to observe the opponent's commitment level to predict which kind they are. To win the focal player must figure out which kind of opponent it is playing against and either best respond by selecting the same choice in all interactions after the first if paired with the first kind of opponent, or apply the cyclic strategy described as the solution to **SC 1** if paired with the second kind of opponent.

SC 5 *Versus gullible opponent.* Here the focal agent must defeat an opposing agent that was trained to best respond to agents playing pure strategies. The opponent should attempt to scout out what strategy the focal agent is playing so it can pick the appropriate counter. To defeat it, the focal

agent should feint toward one resource and then collect the counter to its counter. So for example, if the focal agent successfully feinted that it would pick rock, inducing its opponent to pick paper, the focal agent should then collect and play scissors. This opponent is fairly weak.

- SC 6** *Versus pure rock opponent.* Opponent always plays rock, and commits to it strongly, aiming to collect five resources before interacting. The focal player gets a high score when it picks paper and commits strongly to that choice.
- SC 7** *Versus pure paper opponent.* Same as **SC 6** but opponent plays paper so focal player should play scissors.
- SC 8** *Versus pure scissors opponent.* Same as **SC 6** but opponent plays scissors so focal player should play rock.

Stag Hunt in the matrix: Arena⁴⁸

Individuals collect resources that represent 'hare' (red) or 'stag' (green) and compare inventories in an encounter. Consequences of this inventory comparison are congruent with the classic Stag Hunt matrix game. This game exposes a tension between reward for the group and risk for the individual. The matrix for the interaction is

$$A_{\text{row}} = A_{\text{col}}^T = \begin{bmatrix} 4 & 0 \\ 2 & 2 \end{bmatrix} .$$

Scenarios

- SC 0** *Visiting unconditional stag players.* This is a visitor mode scenario. One focal player joins seven background players. In this case the background population implements the stag policy, it's a good idea for the focal player to join them and play stag.
- SC 1** *Visiting unconditional hare players.* This is a visitor mode scenario. One focal player joins seven background players. In this case the background population implements the

⁴⁸For a video of *Stag Hunt in the matrix: Arena*, see <https://youtu.be/ag0po0MZmzs>

hare policy. Even though the focal player would prefer to play stag, they can't force the others to help them. After determining the others won't change their ways then they should give in and play hare.

SC 2 Focals are resident and visitors are unconditional stag players. This is a resident mode scenario where five focals are joined by three background players. The test is mostly about the focals implementing a cooperative joint policy among themselves. They also must be robust to the presence of the three visitors.

SC 3 Focals are resident and visitors are unconditional hare players. This is a resident mode scenario where five focals are joined by three background players. The test is mostly about the focals implementing a cooperative joint policy among themselves. They also must be robust to the presence of the three visitors.

SC 4 Visiting a population of hair-trigger grim reciprocator bots who initially play stag but, if any partner plays hare once, they give up on trying to cooperate and play hare in all future interactions. This is a visitor mode scenario, the focal player joins seven background players. The focal player would do well by playing stag.

SC 5 Visiting a population of two-strikes grim reciprocator bots who initially play stag but, if their partners play hare twice, they give up on trying to cooperate and play hare in all future interactions. This is like SC4, but this version is a bit more forgiving.

SC 6 Visiting a mixed population of k-strokes grim reciprocator bots with k values from 1 to 3, they initially play stag but, if their partners play hare k times, they then play hare in all future interactions. This is a visitor mode scenario where three focal players meet five background players. The focal players can achieve a high score by sticking with playing stag.

SC 7 Visiting a mixture of pure hare and pure stag players. This is a visitor mode scenario

where three focal players meet five background players. In this case it is useful to scout out specific partners and try to interact preferentially with those who play stag.

Stag Hunt in the matrix: Repeated⁴⁹

See the description of *Stag Hunt in the matrix: Arena*. The difference between the two variants is that *Repeated* is a two-player game while *Arena* is an eight-player game.

Scenarios

SC 0 Partner may play either stag or hare. Here the focal agent plays with a partner sampled from the background population that was trained to play a pure strategy: either to collect mainly stag or mainly hare. However, the specific background player is sampled at test time so it could of either type. All background bots commit strongly to their choice, aiming to collect at least five (or seven) resources before interacting. To get a high score the focal agent should scout out which pure strategy the background bot is playing and then play stag if they play stag and otherwise play hare.

SC 1 Partner typically plays stag. Here the background player always picks stag the best course of action is to join them.

SC 2 Partner typically plays hare. Here the background player always picks hare the best course of action is also to player hare.

SC 3 Partner is a hair-trigger grim reciprocator; i.e. one who initially cooperates but, if defected on once, will retaliate by defecting forever after. The best choice for the focal player is to play stag.

SC 4 Partner is a two-strokes grim reciprocator; i.e. one who initially cooperates, but if defected on twice, will retaliate by defecting forever after. This scenario is similar to SC3 but somewhat more forgiving.

⁴⁹For a video of *Stag Hunt in the matrix: Repeated*, see https://youtu.be/aDp_CArcb1Y

SC 5 *Partner is a tit-for-tat conditional cooperator.*

The focal player should play stag here.

SC 6 *Partner is a tit-for-tat conditional cooperator who occasionally plays hare instead of stag.*

The focal player should play stag here. If employing a conditional cooperation strategy then it's important not to get stuck in a cycle of recrimination as a result of the background player occasionally playing hare. Forgiveness is helpful.

SC 7 *Partner plays stag for a while then switches to hare.* This scenario tests flexibility.

SC 8 *Partner initially plays hare, but if punished, partner then switches to tit-for-tat conditional cooperation.* In this scenario the background player can be encouraged to switch strategies by punishing it.

SC 9 *Partner initially plays hare, but if punished, partner then switches to noisy tit-for-tat conditional cooperation.* This scenario is like SC6 but the background player occasionally plays hare so forgiveness is important.

Reference learning curves

See Figs. [6–54](#).

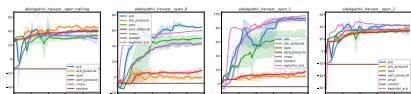


Figure 6 | Allelopathic Harvest: Open

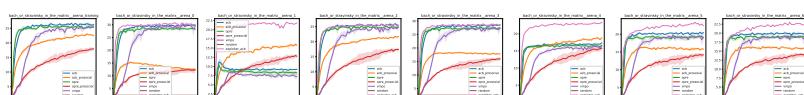


Figure 7 | Bach or Stravinsky in the matrix: Arena

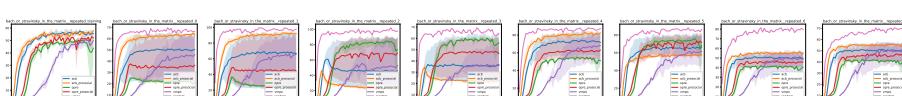


Figure 8 | Bach or Stravinsky in the matrix: Repeated

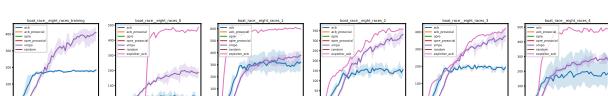


Figure 9 | Boat Race: Eight Races

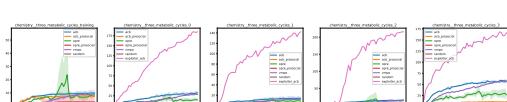


Figure 10 | Chemistry: Three Metabolic Cycles

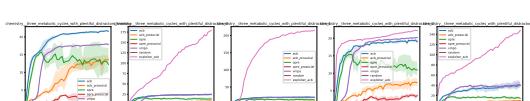


Figure 11 | Chemistry: Three Metabolic Cycles with Plentiful Distractors

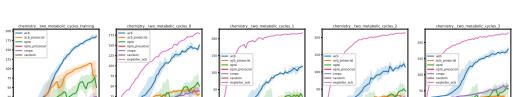


Figure 12 | Chemistry: Two Metabolic Cycles

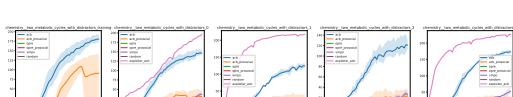


Figure 13 | Chemistry: Two Metabolic Cycles with Distractors

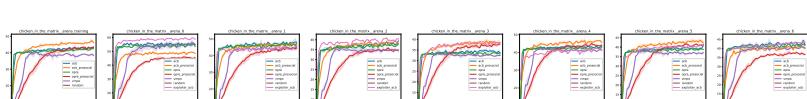


Figure 14 | Chicken in the matrix: Arena

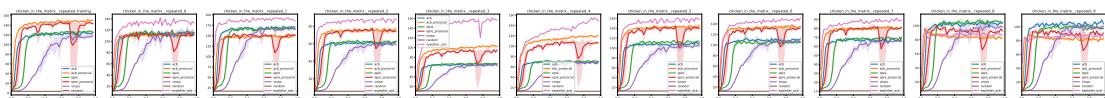


Figure 15 | Chicken in the matrix: Repeated

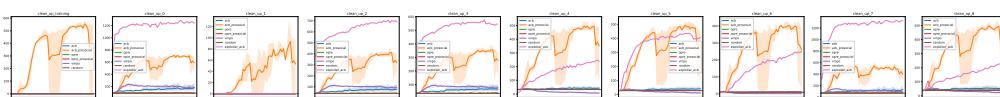


Figure 16 | Clean Up

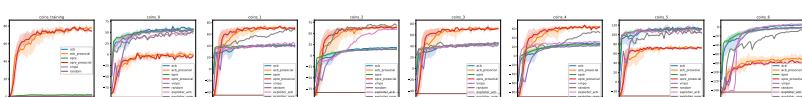


Figure 17 | Coins

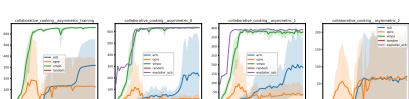


Figure 18 | Collaborative Cooking: Asymmetric

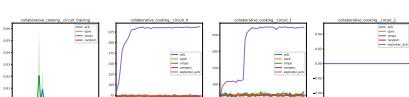


Figure 19 | Collaborative Cooking: Circuit

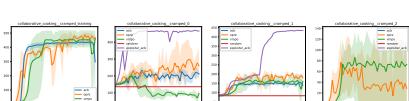


Figure 20 | Collaborative Cooking: Cramped

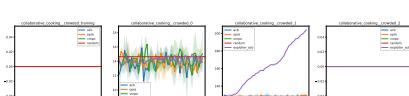


Figure 21 | Collaborative Cooking: Crowded

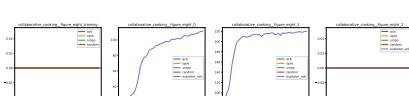


Figure 22 | Collaborative Cooking: Figure Eight

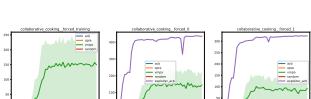


Figure 23 | Collaborative Cooking: Forced

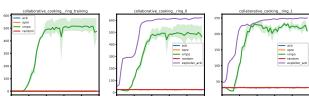


Figure 24 | Collaborative Cooking: Ring

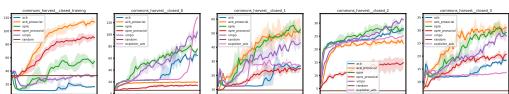


Figure 25 | Commons Harvest: Closed

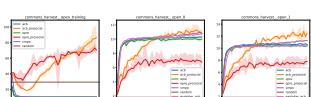


Figure 26 | Commons Harvest: Open

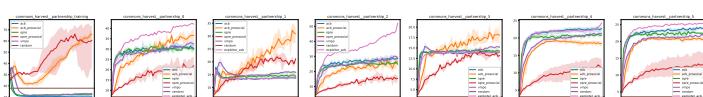


Figure 27 | Commons Harvest: Partnership

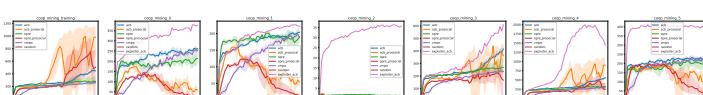


Figure 28 | Coop Mining

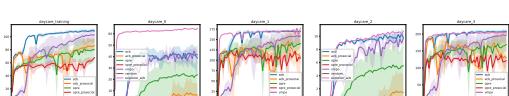


Figure 29 | Daycare

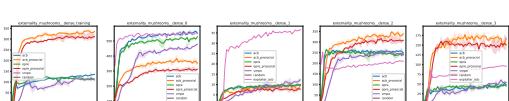


Figure 30 | Externality Mushrooms: Dense

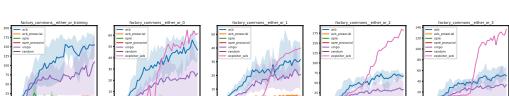


Figure 31 | Factory Commons: Either Or

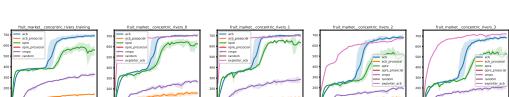


Figure 32 | Fruit Market: Concentric Rivers

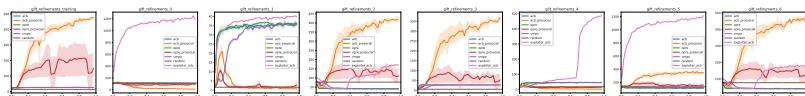


Figure 33 | Gift Refinements

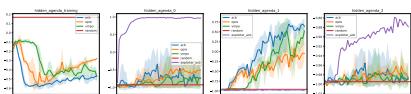


Figure 34 | Hidden Agenda

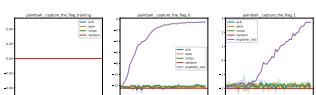


Figure 35 | Paintball: Capture the Flag

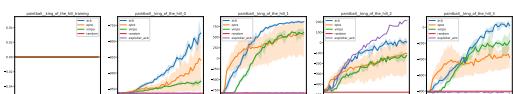


Figure 36 | Paintball: King of the Hill

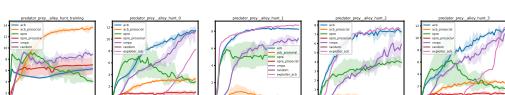


Figure 37 | Predator Prey: Alley Hunt

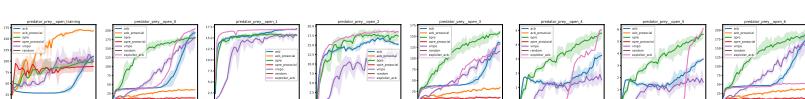


Figure 38 | Predator Prey: Open

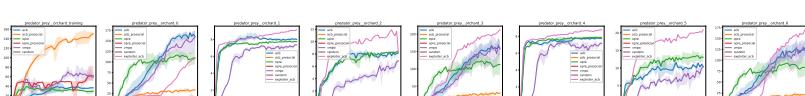


Figure 39 | Predator Prey: Orchard

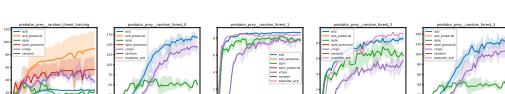


Figure 40 | Predator Prey: Random Forest

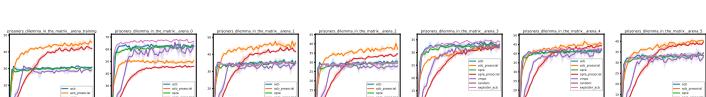


Figure 41 | Prisoners Dilemma in the matrix: Arena

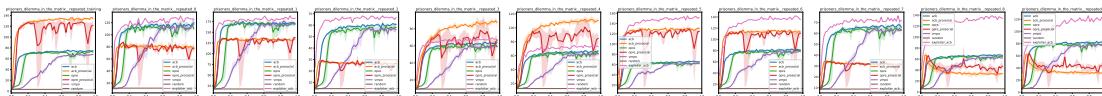


Figure 42 | Prisoners Dilemma in the matrix: Repeated

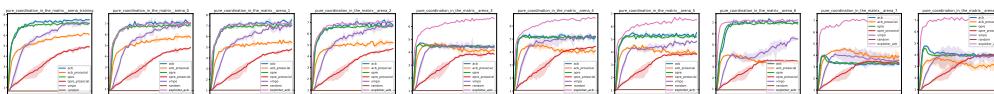


Figure 43 | Pure Coordination in the matrix: Arena

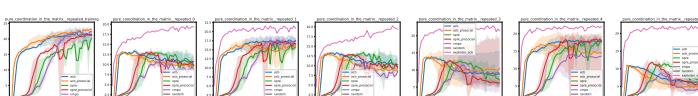


Figure 44 | Pure Coordination in the matrix: Repeated

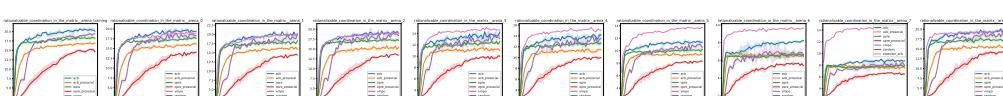


Figure 45 | Rationalizable Coordination in the matrix: Arena

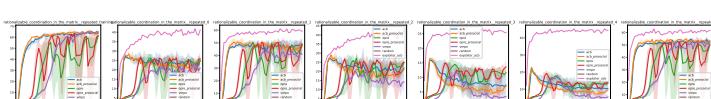


Figure 46 | Rationalizable Coordination in the matrix: Repeated

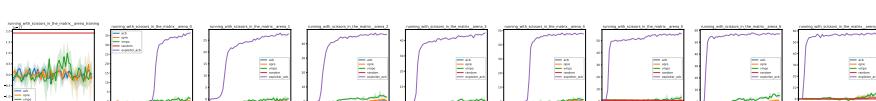


Figure 47 | Running With Scissors in the matrix: Arena

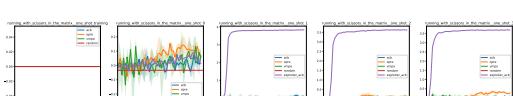


Figure 48 | Running With Scissors in the matrix: One Shot

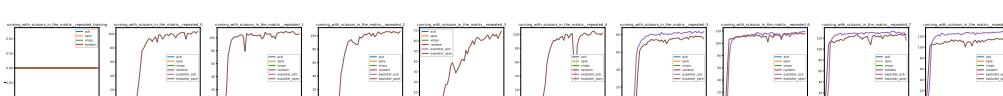


Figure 49 | Running With Scissors in the matrix: Repeated

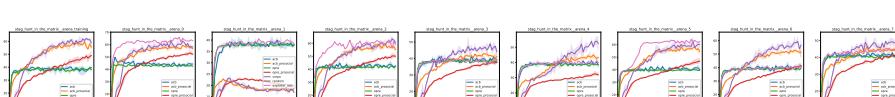


Figure 50 | Stag Hunt in the matrix: Arena

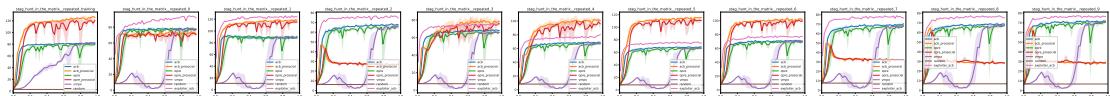


Figure 51 | Stag Hunt in the matrix: Repeated

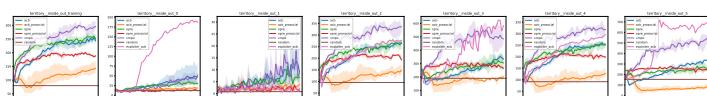


Figure 52 | Territory: Inside Out

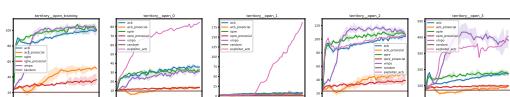


Figure 53 | Territory: Open

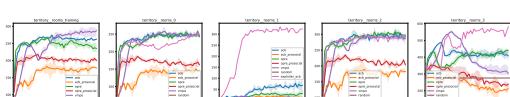


Figure 54 | Territory: Rooms

References

- D. Acemoglu. Harms of AI. Technical report, National Bureau of Economic Research, 2021.
- D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.
- R. Axelrod. *The Evolution of Cooperation*. Basic Books, 1984.
- B. Baker, I. Kanitscheider, T. Markov, Y. Wu, G. Powell, B. McGrew, and I. Mordatch. Emergent tool use from multi-agent autocurricula. In *International Conference on Learning Representations*, 2019.
- M. A. Bakker, R. Everett, L. Weidinger, I. Gabriel, W. S. Isaac, J. Z. Leibo, and E. Hughes. Modelling cooperation in network games with spatio-temporal complexity. *arXiv preprint arXiv:2102.06911*, 2021.
- D. Balduzzi, M. Garnelo, Y. Bachrach, W. Czarnecki, J. Perolat, M. Jaderberg, and T. Graepel. Open-ended learning in symmetric zero-sum games. In *International Conference on Machine Learning*, pages 434–443, 2019.
- A. Barreto, D. Borsa, S. Hou, G. Comanici, E. Aygün, P. Hamel, D. Toyama, S. Mourad, D. Silver, D. Precup, et al. The option keyboard: Combining skills in reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 32:13052–13062, 2019.
- C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Józefowicz, S. Gray, C. Olsson, J. Pachocki, M. Petrov, H. P. de Oliveira Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, I. Sutskever, J. Tang, F. Wolski, and S. Zhang. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.
- R. Burnell, W. Schellaert, J. Burden, T. D. Ullman, F. Martinez-Plumed, J. B. Tenenbaum, D. Rutar, L. G. Cheke, J. Sohl-Dickstein, M. Mitchell, D. Kiela, M. Shanahan, E. M. Voorhees, A. G. Cohn, J. Z. Leibo, and J. Hernandez-Orallo. Rethink reporting of evaluation results in ai. *Science*, 380(6641):136–138, 2023.
- M. Carroll, R. Shah, M. K. Ho, T. Griffiths, S. Sebia, P. Abbeel, and A. Dragan. On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems*, 32, 2019.
- C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI/IAAI*, 1998(746-752):2, 1998.
- K. Cobbe, O. Klimov, C. Hesse, T. Kim, and J. Schulman. Quantifying generalization in reinforcement learning. In *International Conference on Machine Learning*, pages 1282–1289. PMLR, 2019.
- M. Crosby, B. Beyret, and M. Halina. The animal-ai olympics. *Nature Machine Intelligence*, 1(5):257–257, 2019.
- A. Dafoe, E. Hughes, Y. Bachrach, T. Collins, K. R. McKee, J. Z. Leibo, K. Larson, and T. Graepel. Open problems in cooperative AI. *arXiv preprint arXiv:2012.08630*, 2020.
- E. A. Duéñez-Guzmán, K. R. McKee, Y. Mao, B. Coppin, S. Chiappa, A. S. Vezhnevets, M. A. Bakker, Y. Bachrach, S. Sadedin, W. Isaac, et al. Statistical discrimination in learning agents. *arXiv preprint arXiv:2110.11404*, 2021.
- L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, pages 1407–1416. PMLR, 2018.
- J. Farebrother, M. C. Machado, and M. Bowling. Generalization and regularization in DQN. *arXiv preprint arXiv:1810.00123*, 2018.
- J. Foerster, R. Y. Chen, M. Al-Shedivat, S. Whiteson, P. Abbeel, and I. Mordatch. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 122–130, 2018.

- D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan. Cooperative inverse reinforcement learning. *Advances in neural information processing systems*, 29, 2016.
- A.-B. Haidich. Meta-analysis in medical research. *Hippokratia*, 14(Suppl 1):29, 2010.
- T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.
- J. Henrich and M. Muthukrishna. The origins and psychology of human cooperation. *Annual Review of Psychology*, 72:207–240, 2021.
- J. Hernández-Orallo. Evaluation in artificial intelligence: from task-oriented to ability-oriented measurement. *Artificial Intelligence Review*, 48: 397–447, 2017.
- M. Hessel, H. Soyer, L. Espeholt, W. Czarnecki, S. Schmitt, and H. van Hasselt. Multi-task deep reinforcement learning with popart. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3796–3803, 2019.
- M. Hessel, M. Kroiss, A. Clark, I. Kemaev, J. Quan, T. Keck, F. Viola, and H. van Hasselt. Podracer architectures for scalable reinforcement learning. *arXiv preprint arXiv:2104.06272*, 2021.
- E. Hughes, J. Z. Leibo, M. G. Philips, K. Tuyls, E. A. Duéñez-Guzmán, A. G. Castañeda, I. Dunning, T. Zhu, K. R. McKee, R. Koster, H. Roff, and T. Graepel. Inequity aversion improves cooperation in intertemporal social dilemmas. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3330–3340. 2018.
- D. Hume. *A treatise of human nature*. Courier Corporation, 1739/2012.
- M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.
- M. Jaderberg, W. M. Czarnecki, I. Dunning, L. Marris, G. Lever, A. G. Castaneda, C. Beattie, N. C. Rabinowitz, A. S. Morcos, A. Ruderman, et al. Human-level performance in 3d multi-player games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019.
- M. A. Janssen, R. Holahan, A. Lee, and E. Ostrom. Lab experiments for the study of social-ecological systems. *Science*, 328(5978):613–617, 2010.
- M. B. Johanson, E. Hughes, F. Timbers, and J. Z. Leibo. Emergent bartering behaviour in multi-agent reinforcement learning. *arXiv preprint arXiv:2205.06760*, 2022.
- P. Koehn and R. Knowles. Six challenges for neural machine translation. In *First Workshop on Neural Machine Translation*, pages 28–39. Association for Computational Linguistics, 2017.
- R. Köster, K. R. McKee, R. Everett, L. Weidinger, W. S. Isaac, E. Hughes, E. A. Duéñez-Guzmán, T. Graepel, M. Botvinick, and J. Z. Leibo. Model-free conventions in multi-agent reinforcement learning with heterogeneous preferences. *arXiv preprint arXiv:2010.09054*, 2020.
- R. Köster, D. Hadfield-Menell, R. Everett, L. Weidinger, G. K. Hadfield, and J. Z. Leibo. Spurious normativity enhances learning of compliance and enforcement behavior in artificial agents. *Proceedings of the National Academy of Sciences*, 119(3):e2106028118, 2022.
- M. Lanctot, V. Zambaldi, A. Gruslys, A. Lazaridou, K. Tuyls, J. Pérolat, D. Silver, and T. Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in neural information processing systems (NeurIPS)*, pages 4190–4203, 2017.
- J. Z. Leibo, V. Zambaldi, M. Lanctot, J. Marecki, and T. Graepel. Multi-agent Reinforcement Learning in Sequential Social Dilemmas. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AA-MAS 2017)*, São Paulo, Brazil, 2017.
- J. Z. Leibo, J. Perolat, E. Hughes, S. Wheelwright, A. H. Marblestone, E. Duéñez-Guzmán, P. Sunehag, I. Dunning, and T. Graepel. Malthusian reinforcement learning. In *Proceedings of the*

- 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1099–1107, 2019.
- J. Z. Leibo, E. A. Dueñez-Guzman, A. Vezhnevets, J. P. Agapiou, P. Sunehag, R. Koster, J. Matyas, C. Beattie, I. Mordatch, and T. Graepel. Scalable evaluation of multi-agent reinforcement learning with Melting Pot. In *International Conference on Machine Learning*, pages 6187–6199. PMLR, 2021.
- A. Lerer and A. Peysakhovich. Maintaining cooperation in complex social dilemmas using deep reinforcement learning. *arXiv preprint arXiv:1707.01068*, 2017.
- R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *arXiv preprint arXiv:1706.02275*, 2017.
- R. D. Luce and H. Raiffa. *Games and decisions: Introduction and critical survey*. Courier Corporation, 1957/1989.
- F. Martínez-Plumed, R. B. Prudêncio, A. Martínez-Usó, and J. Hernández-Orallo. Item response theory in AI: Analysing machine learning classifiers at the instance level. *Artificial intelligence*, 271:18–42, 2019.
- F. Martínez-Plumed, J. Hernández-Orallo, and E. Gómez. Tracking the impact and evolution of ai: The aicollaboratory. 2020.
- K. R. McKee, I. Gemp, B. McWilliams, E. A. Duèñez-Guzmán, E. Hughes, and J. Z. Leibo. Social diversity and social preferences in mixed-motive reinforcement learning. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pages 869–877, 2020.
- N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.
- V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.
- Z. Obermeyer, B. Powers, C. Vogeli, and S. Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019.
- A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- E. Ostrom. *Understanding institutional diversity*. Princeton university press, 2009.
- S. Ott, A. Barbosa-Silva, K. Blagec, J. Brauner, and M. Samwald. Mapping global dynamics of benchmark creation and saturation in artificial intelligence. *Nature Communications*, 13(1):1–11, 2022.
- R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR, 2013.
- J. Perolat, J. Z. Leibo, V. Zambaldi, C. Beattie, K. Tuyls, and T. Graepel. A multi-agent reinforcement learning model of common-pool resource appropriation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3643–3652, 2017.
- A. Peysakhovich and A. Lerer. Prosocial learning agents solve generalized stag hunts better than selfish ones. *arXiv preprint arXiv:1709.02865*, 2017.
- I. Rahwan, M. Cebrian, N. Obradovich, J. Bongard, J.-F. Bonnefon, C. Breazeal, J. W. Crandall, N. A. Christakis, I. D. Couzin, M. O. Jackson, et al. Machine behaviour. *Nature*, 568(7753):477–486, 2019.
- T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 4295–4304. PMLR, 2018.

- R. Roelofs, V. Shankar, B. Recht, S. Fridovich-Keil, M. Hardt, J. Miller, and L. Schmidt. A meta-analysis of overfitting in machine learning. *Advances in Neural Information Processing Systems*, 32, 2019.
- O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.
- S. Russell. *Human compatible: Artificial intelligence and the problem of control*. Penguin, 2019.
- S. Russell, D. Dewey, and M. Tegmark. Research priorities for robust and beneficial artificial intelligence. *Ai Magazine*, 36(4):105–114, 2015.
- D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529 (7587):484, 2016.
- D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.
- D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.
- N. Soares and B. Fallenstein. Aligning superintelligence with human interests: A technical research agenda. *Machine Intelligence Research Institute (MIRI) technical report*, 8, 2014.
- H. F. Song, A. Abdolmaleki, J. T. Springenberg, A. Clark, H. Soyer, J. W. Rae, S. Noury, A. Ahuja, S. Liu, D. Tirumala, N. Heess, D. Belov, M. Riedmiller, and M. M. Botvinick. V-mpo: On-policy maximum a posteriori policy optimization for discrete and continuous control. In *International Conference on Learning Representations*, 2020.
- D. Strouse, K. McKee, M. Botvinick, E. Hughes, and R. Everett. Collaborating with humans without human data. *Advances in Neural Information Processing Systems*, 34:14502–14515, 2021.
- P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2085–2087, 2018.
- R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- R. S. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.
- R. S. Sutton, J. Modayil, M. Delp, T. Degris, P. M. Pilarski, A. White, and D. Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 761–768, 2011.
- L. Tesfatsion. Agent-based computational economics: Overview and brief history. Working Paper 21004, Department of Economics, Iowa State University, 2021.
- A. Vezhnevets, Y. Wu, M. Eckstein, R. Leblond, and J. Z. Leibo. Options as responses: Grounding behavioural hierarchies in multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 9733–9742. PMLR, 2020.
- A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*, pages 3540–3549. PMLR, 2017.
- E. Vinitsky, R. Köster, J. P. Agapiou, E. A. Duéñez-Guzmán, A. S. Vezhnevets, and J. Z. Leibo. A learning agent that acquires social norms

from public sanctions in decentralized multi-agent settings. *Collective Intelligence*, 2(2):26339137231162025, 2023.

- O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, et al. Grandmaster level in starcraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.
- R. E. Wang, S. A. Wu, J. A. Evans, J. B. Tenenbaum, D. C. Parkes, and M. Kleiman-Weiner. Too many cooks: Coordinating multi-agent collaboration through inverse planning. *arXiv preprint arXiv:2003.11778*, 2020.
- J. W. Weibull. *Evolutionary game theory*. MIT press, 1997.
- L. Weidinger, J. Mellor, M. Rauh, C. Griffin, J. Uesato, P.-S. Huang, M. Cheng, M. Glaese, B. Balle, A. Kasirzadeh, et al. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*, 2021.
- D. Ye, G. Chen, W. Zhang, S. Chen, B. Yuan, B. Liu, J. Chen, Z. Liu, F. Qiu, H. Yu, et al. Towards playing full moba games with deep reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.
- A. Zhang, N. Ballas, and J. Pineau. A dissection of overfitting and generalization in continuous reinforcement learning. *arXiv preprint arXiv:1806.07937*, 2018.