The Hive Mind is a Single Reinforcement Learning Agent.

Karthik Soma[1,2†], Yann Bouteiller[1,2†], Heiko Hamann[3], Giovanni Beltrame[1,2]

[1] Polytechnique Montreal .
[2] Mila - Quebec AI Institute .
[3] University of Konstanz .

[†]These authors contributed equally to this work.

Abstract

Decision-making is an essential attribute of any intelligent agent or group. Natural systems are known to converge to optimal strategies through at least two distinct mechanisms: collective decision-making via imitation of others, and individual trial-and-error. This paper establishes an equivalence between these two paradigms by drawing from the well-established collective decision-making model of nest-hunting in swarms of honey bees. We show that the emergent distributed cognition (sometimes referred to as the *hive mind*) arising from individual bees following simple, local imitation-based rules is that of a single online reinforcement learning (RL) agent interacting with many parallel environments. The update rule through which this macro-agent learns is a bandit algorithm that we coin *Maynard-Cross Learning*. Our analysis implies that a group of cognition-limited organisms can be equivalent to a more complex, reinforcement-enabled entity, substantiating the idea that group-level intelligence may explain how seemingly simple and blind individual behaviors are selected in nature.
From a biological perspective, this analysis suggests how such imitation strategies evolved: they constitute a scalable form of reinforcement learning at the group level, aligning with theories of kin and group selection. Beyond biology, the framework offers new tools for analyzing economic and social systems where individuals imitate successful strategies, effectively participating in a collective learning process. In swarm intelligence, our findings will inform the design of scalable collective systems in artificial domains, enabling RL-inspired mechanisms for coordination and adaptability at scale.

Keywords: Swarm Intelligence, Reinforcement Learning, Evolutionary Game Theory, Opinion Dynamics

1 Introduction

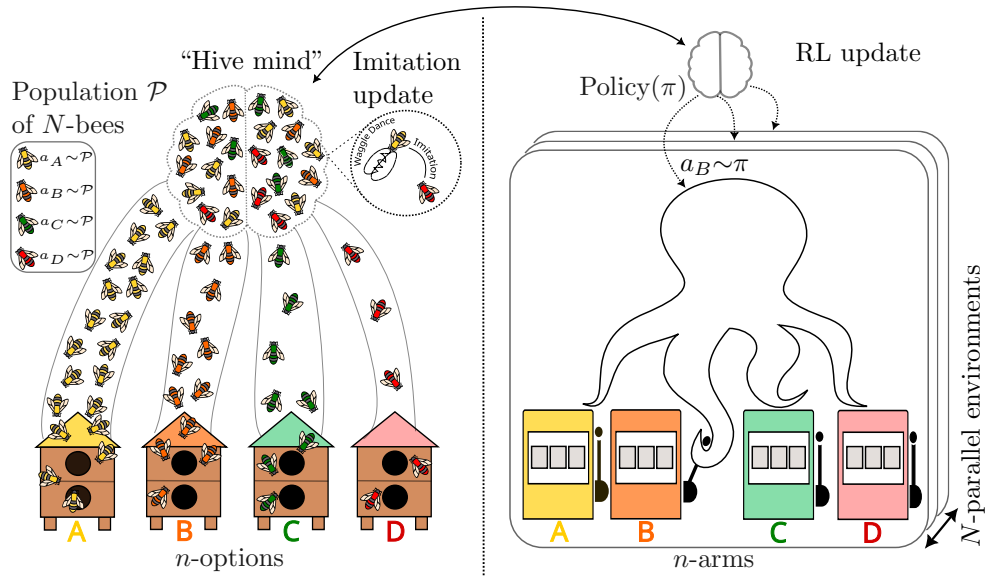


Fig. 1: The "hive mind" of a swarm of N bees nest-hunting among n options is a single n-armed bandit RL agent learning from N environments in parallel.

Decision-making is the ability to choose the best action amongst all available options in a given scenario. Defining what a "best action" means requires ranking all possible outcomes [1]. Most often, the preference among outcomes is abstracted out as a single scalar signal [2] conveyed to the agent/individual[1] by the environment as feedback for making a decision, which casts optimal decision-making into a maximization problem.

In nature, at least two distinct mechanisms allow single agents or groups of individuals to converge towards making optimal decisions. The first of these mechanisms adopts an individualistic approach, whereby the agent *learns* to make optimal decisions through trial and error. This paradigm is known as Reinforcement Learning (RL), where the agent learns a policy that maximizes the rewards it receives upon interacting with its environment [3]. Numerous studies in computational neuroscience have established links between learning processes happening in the brains of living beings and the formal framework of algorithmic RL [4, 5]. In this paper, we are specifically interested in the multi-armed bandit framework [3], where a single RL agent learns to make choices among different options (or "arms") in an online fashion. Online RL agents learn by either interacting with a single environment (a setting recently referred to as the "streaming" setting [6, 7]), or, when possible, by collecting samples

[1] To avoid confusion, we use "agent" in the context of RL and "individual" in the context of a population.

in parallel from multiple copies of the same environment simultaneously (the "parallel" setting [8]). To stabilize RL, the learning rate and the number of parallel environments are crucial in the streaming and parallel cases, respectively. Among the many learning algorithms designed to solve multi-armed bandits (Upper-Confidence-Bound [9], ϵ-greedy [3], Gradient Bandit [10], etc.), we consider the Cross Learning (CL) [11] update rule (named after the economist John G. Cross [12] and not to be mixed with more recent overlapping nomenclature). CL is closely related to the Gradient Bandit algorithm, an RL algorithm directly based on the policy gradient theorem [3]. The policy gradient theorem is also the basis of many more advanced policy gradient (e.g., REINFORCE [13] and PPO [14]) and actor-critic (e.g., SAC [15]) RL algorithms.

The second mechanism takes a collective approach, where individuals mimic other individuals in the group to make decisions. Typically, this process is cast as a consensus problem, in which all individuals in a population have to agree on the best decision from a set of alternatives. This problem is commonly referred to as best-of-n collective decision-making (CDM). CDM arises in various domains, such as honey bee colonies [16, 17], human societies [18], and robot swarms [19, 20]. *Imitation of success* [21], from Evolutionary Game Theory (EGT), is a solution to CDM where individuals mimic randomly chosen neighbors based on how successful those neighbors' decisions have been. Another solution to CDM is the *weighted voter model* [20, 22], which comes from modeling the nest-hunting behavior of honey bees [23]. During nest-hunting, honey bees are known to adopt the behavior represented in Fig. 1 (left-hand half): after scouting one of n potential nesting areas, bees come back to the initial location of the swarm and perform a "waggle dance" [23] that describes the coordinates of the option they have explored. This dance is performed at a frequency that is proportional to the estimated quality of the explored area. Other bees go scout the area corresponding to the first dance they witness, and this process repeats until the colony reaches a quorum [24]. At this point, the entire swarm takes off and leaves for the winning site (note this is a simplified description of bees' behavior, see [24–27] for details). We refer to this distributed behavior as that of a "hive mind", motivated by two factors: (1) biologists have already witnessed parallels between the emergent distributed cognition arising from individual-to-individual interactions and cognition in neuron-based complex brains of vertebrates [27], and (2): we formally show that the bee colony collectively learns as if it were one single RL entity.

In this paper, we highlight how imitation mechanisms in groups and individual trial-and-error mechanisms can be bridged via different variants of the Replicator Dynamic [28] (RD). In particular, we show that the model used by biologists to describe the nest-hunting behavior of honey bees amounts to a single-agent RL process at the swarm level, as illustrated in Fig. 1. To the best of our knowledge, this is the first work to formally establish this equivalence. Specifically, our contributions are:

- We first remark that a large population of non-learning individuals following the *imitation of success model* can equivalently be seen as a single abstract RL agent following the *Cross Learning* update rule.
- We then show that the *weighted voter model* used to model the behavior of individual honey bees during nest-hunting also aggregates to a single-agent RL algorithm at the macro-organism level, that we coin *Maynard-Cross Learning*.

- We extend our theoretical analysis with further results from simulation.

The first two contributions, which detail how social imitation models aggregate into a single reinforcement learning agent, provide a formal description of diverse instances of collective intelligence.

2 Background

2.1 Multi-armed bandits and Cross Learning (J.G. Cross)

Multi-armed bandits (Fig. 1, right-hand half) are the simplest type of environment encountered in RL literature. They consist of a discrete set of available actions, called "arms", among which the agent has to find the most rewarding. In the n-armed bandits considered in this paper, pulling an arm $a \in \{1, \ldots, n\}$ returns a real-valued reward $r_a \in [0, 1]$ sampled from a hidden distribution $r(a)$. The objective for an RL agent playing a multi-armed bandit is to learn a policy, denoted by the probability vector $\pi = (\pi_1, \ldots . \pi_n)$, that maximizes the rewards obtained upon pulling the arms. Different optimization strategies exist to find such policies, one of the oldest being Cross Learning [11]:

Definition 1. Let k be an action and r_k a corresponding reward sample ($r_k \sim r(k)$). Let π_a denote the a^{th} component of π. Cross Learning (CL) updates the policy π as:

$$\forall a, \pi_a \leftarrow \pi_a + r_k \times \begin{cases} 1 - \pi_a & \text{if } a = k \\ -\pi_a & \text{otherwise} \end{cases} \tag{1}$$

For convenience, when sampling reward r_k from action k, we denote the expected policy update on action a's probability π_a as:

$$d\pi_a(k) = \mathbb{E}_{r_k \sim r(k)}[r_k] \times \begin{cases} 1 - \pi_a & \text{if } a = k \\ -\pi_a & \text{otherwise} \end{cases} \tag{2}$$

In CL, every reward r_k sampled by applying an action k directly affects the probabilities of all actions. As noted earlier, CL is close to the Gradient Bandit algorithm, which performs a similar update at the parameter level (called "preferences") of a parametric policy rather than directly updating the probability vector.

2.2 Evolutionary Game Theory

Evolutionary Game Theory (EGT) is the study of population games [21]. A population $\mathcal{P}$ is made of a large number of individuals, where any individual i is associated with a *type*, denoted by $T_i \in \{1, \ldots, n\}$. The *population vector* $\pi = (\pi_1, \ldots, \pi_n)$ represents the fraction of individuals in each type ($\sum_i \pi_i = 1$). Individuals are repeatedly paired at random to play a game, each receiving a separate payoff. Individuals adapt their type based on their payoff according to a revision protocol.

Remark 1. Population-policy equivalence. For the argument of our paper, it is interesting to interchangeably define $\pi = (\pi_1, \ldots, \pi_n)$ both as a multi-armed bandit RL policy and as a population vector [12]. This is possible because in both cases the vector π is constrained to the probability simplex. Further, note that uniformly sampling an individual of type a from the population $\mathcal{P}$ (represented by the population vector π) is equivalent to sampling an action a from the policy π.

2.2.1 Imitation of success and the Taylor Replicator Dynamic

In evolutionary dynamics, an important revision protocol is *imitation of success* [21]:

Definition 2. In the "imitation of success" revision model R_{success}, any individual $i \in \mathcal{P}$ of type $T_i = a$ executes the following process:

- i samples a random individual $j \sim \mathcal{U}(\mathcal{P})$ to *imitate*. Let T_j be b.
- Both individuals i and j play a 2-player game in which they receive payoffs r_a and r_b respectively ($0 \leq r_{a,b} \leq 1$). Each payoff depends on the types of both individuals.
- i switches from type a to type b with probability r_b.

One can easily see why this rule is called "imitation of success": i imitates j based on j's payoff. Imitation of success is typically used to model replication of the fittest in evolution for biology, or certain human behaviors in economics [29]. When aggregated across the population, this revision model yields a famous evolutionary dynamic known as the *Taylor Replicator Dynamic* [28, 30] (TRD) (see Lemma 1):

$$\dot{\pi}_a = \pi_a(q_a^\pi - v^\pi), \tag{3}$$

where $\dot{\pi}_a$ is the derivative of the a-th component of the population vector, $q_a^\pi := \mathbb{E}[r_a]$ is the expected payoff of the type a against the current population, and $v^\pi := \sum_b \pi_b \mathbb{E}[r_b]$ is the current average payoff of the entire population.

2.2.2 Weighted Voter Model and the Maynard-Smith Replicator Dynamic

The *weighted voter* model [22] instead typically models the nest-hunting behavior of honey bees described in Section 1 [19, 23]:

Definition 3. In the "weighted voter" revision model R_{wvoter}, any bee $i \in \mathcal{P}$ of type $T_i = a$ (where a corresponds to a nest-site option) executes the following process:

- i estimates the quality of its current type $r_a \sim r(a)$, where $0 \leq r_a \leq 1$.
- After obtaining r_a, i locally broadcasts its type at a frequency proportional to r_a.
- i switches its type to the first type b that it perceives from its neighborhood. Assuming all individuals are well mixed in the population [31], the corresponding expected probability of i switching to type b is the proportion of votes cast for b within its neighborhood:

$$P^{(i)}_{\text{neighborhood}}(b \leftarrow a) = \frac{N^{(i)}_b \mathbb{E}[r_b]}{\sum_l N^{(i)}_l \mathbb{E}[r_l]}$$

  where $N^{(i)}_k$ is the number of individuals of type k in the neighborhood of i.

Note that, in this model, honey bees do not need to directly observe the quality estimate of other scout bees, but only their type. In Section 3.2, we show that the weighted voter revision model aggregates to a variant of the TRD, called the *Maynard-Smith Replicator Dynamic* [32] (MRD):

$$\dot{\pi}_a = \frac{\pi_a}{v^\pi}(q^\pi_a - v^\pi) \tag{4}$$

3 Methodology

In Section 3.1, we remark that previous results from the literature yield an interesting insight binding R_{success} and reinforcement learning together: a large population following the "imitation of success" revision model can equivalently be considered as a single RL agent. While this is a direct consequence of previously known results, we could not find this insight formulated in the literature and thus we took the liberty to formalize it as Proposition 1. Then, in Section 3.2, we prove that the weighted voter revision model also aggregates to an RL algorithm. In other words, our analysis indicates that, at least in the well-studied case of nest-hunting, a swarm of honeybees collectively acts as a single RL entity. We formalize this result as Proposition 2.

3.1 Imitation of Success and Cross Learning

Evolutionary Game Theorists have long been interested in the "imitation of success" revision protocol, as it models replication of the fittest in the evolutionary setting. In this literature, it is famously known that a population of individuals following R_{success} aggregates to the Taylor Replicator Dynamic (see for instance [21, 28] and the proof in the Appendix):

Lemma 1. An infinite population of individuals adopting R_{success} follows the TRD:

$$d\pi_a = \pi_a(q^\pi_a - v^\pi), \tag{5}$$

where π_a is the proportion of type a in the population, q_a^π is the expected payoff (also called "fitness") of type a against the population, and v^π is the average fitness of the population.

Our choice of notation in Lemma 1 is reminiscent of the RL literature and may seem unusual from an EGT perspective. In fact, this choice is motivated by another, lesser-known result from [12, 33], who showed that the Cross Learning RL algorithm performs updates that are also similar to the TRD (proof in Appendix):

Lemma 2. In expectation, an RL agent learning via the CL update rule follows:

$$\mathbb{E}[d\pi_a] = \pi_a(q_a^\pi - v^\pi), \tag{6}$$

where q_a^π is the action-value of a, and v^π is the value of policy π.

With this notation, it is straightforward to combine Lemmas 1 and 2, provided a sensible duality exists between the identical terms. Remember how Remark 1 describes a duality between the population vector of Lemma 1 and the policy of Lemma 2, π. In Lemma 1, the fitness q_a^π is the payoff that an individual of type a can expect on average when encountering a random individual from the population π. In Lemma 2, the dual of this individual of type a is an action a sampled from the policy π, whose expected reward is the action-value q_a^π. Similarly, the dual of the population fitness v^π is the policy-value. In other words, under R_{success}, individuals sampled from the population π can equivalently be regarded as action samples from an RL macro-agent, whose rewards are the individuals' payoffs. The agent maximizes the average payoff of the group via Exact Cross Learning (we call "exact" the RL algorithm that directly applies expected updates instead of updates computed from sample estimates):

Proposition 1. An infinite population of individuals following R_{success} can equivalently be seen as an RL agent following Exact Cross Learning, i.e.,

$$d^{\text{success}}\pi_a = \mathbb{E}[d^{\text{CL}}\pi_a] \,, \tag{7}$$

where π is both a population vector and a vector of action-probabilities, $d^{\text{success}}\pi$ is the single-step change in the population vector π under the "imitation of success" revision model , and $d^{\text{CL}}\pi$ is an update performed by CL on the policy π.

Proof Direct consequence of Lemmas 1 and 2. □

Intuitively, this macroscopic RL process can appear as a mere by-product of individuals in the population imitating others to optimize for their own success. But let us now turn our attention to organisms that may not even have a notion of individual success to optimize: honey bees.

3.2 Weighted Voters and Maynard-Cross Learning

We now consider a large population of $N \gg 1$ honey bees with average local neighborhood size $M \gg 1$, seeking agreement on which nesting site to select by applying R_{wvoter}. We show that, although R_{wvoter} is a blind imitation protocol where individual bees simply imitate the first type they encounter, a swarm of bees following R_{wvoter} aggregates to an RL agent at the macro-organism level. In the R_{wvoter} model, individual bees are blind imitators, but active promoters. Each bee i of type $T_i = k$ and payoff sample $r_k^{(i)} \sim r(k)$ has a tangible stochastic influence on the local expected inflow of other bees that it rallies to its own type k within its local neighborhood $\mathcal{N}_i$:

$$P^{(i)}_{\text{neighborhood}}(k \leftarrow \cdot) = \frac{r_k^{(i)}}{\sum\limits_{j \in \mathcal{N}_i} r^{(j)}}, \tag{8}$$

where $r_k^{(i)}$ represents i's broadcasting frequency, and $\sum_{j \in \mathcal{N}_i} r^{(j)}$ represents the total broadcasting frequency of i's local neighborhood. The expected outflow attributable to i on the entire swarm from any type $a \neq k$ to the type k is thus:

$$P^{(i)}(k \leftarrow a) = \frac{N_a^{(i)}}{N} \frac{r_k^{(i)}}{\sum\limits_{j \in \mathcal{N}_i} r^{(j)}}, \tag{9}$$

where $N_a^{(i)}$ is the number of type-a bees within i's neighborhood. For simplicity, we assume that M (and thus also $N^{(i)}$) is large. Assuming that all individuals are well mixed in the population [31], it follows that $\frac{N_a^{(i)}}{M} = \pi_a$:

$$\begin{aligned} P^{(i)}(k \leftarrow a) &= \frac{M\pi_a}{N} \frac{r_k^{(i)}}{\sum\limits_{j \in \mathcal{N}_i} r^{(j)}} \\ &= \frac{\pi_a}{N} \frac{r_k}{\frac{1}{M} \sum\limits_{j \in \mathcal{N}_i} r^{(j)}} \\ &= \alpha \frac{r_k}{v^\pi} \pi_a, \end{aligned} \tag{10}$$

where v^π is the average population payoff and $\alpha := \frac{1}{N}$. Summing over all types except k (whose outflow to k is 0), we obtain the total inflow into type k attributable to i:

$$\begin{aligned} \sum_{a \neq k} P^{(i)}(k \leftarrow a) &= \sum_{a \neq k} \alpha \frac{r_k}{v^\pi} \pi_a \\ &= \alpha \frac{r_k}{v^\pi} (1 - \pi_k) \, . \end{aligned} \tag{11}$$

Given a bee i of type k, Eq. (10) describes the outflow that its waggle dance induces within the swarm from any type a to the type k, while Eq. (11) describes the corresponding inflow into type k from all types. In other words, the bee i can be seen as an action-reward sample whose influence on the population vector π is described by an RL update rule that we coin *α-Maynard-Cross Learning* (α-MCL):

Definition 4. Let k be an action and $r_k \sim r(k)$ a corresponding reward sample. MCL updates the policy π as:

$$\forall a, \pi_a \leftarrow \pi_a + \alpha \frac{r_k}{v^\pi} \begin{cases} 1 - \pi_a & \text{if } a = k \\ -\pi_a & \text{otherwise} \end{cases} \tag{12}$$

where v^π is the current value of policy π.

Here, the meaning of α is of peculiar interest. In Eq. (12), α looks very much like a learning rate, i.e., a hyperparameter that RL practitioners typically set to a small value in order to downsize the amplitude of individual policy updates. When dealing with on-policy RL algorithms, computer scientists often evaluate a policy in several parallel copies of the same environment [34], average the corresponding updates into a single policy update, and downsize this update via a small learning rate. The common rule of thumb is that the learning rate can be larger when there are enough parallel environments. But in Definition 4, α instead has a clear population-based meaning: it is the inverse of the number of bees acting in parallel in the swarm, as we are describing the influence of one single bee on the population. Crucially, this influence is that of an RL agent, but the individual bee itself is in no way an RL agent: it only blindly imitates its peers! Instead, everything happens as if nature had evolved individual bees' behavior such that it aggregates to an RL agent at the macro-organism level: the "hive mind", whose rewards are the quality estimates of nest site samples. Summing the influence of all individual bees yields:

$$\begin{aligned} d\pi_a &= \sum_{i=0}^{N} \frac{r^{(i)}}{N v^\pi} \begin{cases} 1 - \pi_a & \text{if } a = k \\ -\pi_a & \text{otherwise} \end{cases} \\ &= \sum_k \frac{N_k q_k^\pi}{N v^\pi} \begin{cases} 1 - \pi_a & \text{if } a = k \\ -\pi_a & \text{otherwise} \end{cases} \\ &= \frac{1}{N v^\pi} (N_a q_a^\pi (1 - \pi_a) - \sum_{k \neq a} N_k q_k^\pi \pi_a) \\ &= \frac{1}{N v^\pi} (N_a q_a^\pi - \sum_k N_k q_k^\pi \pi_a) \\ &= \frac{1}{v^\pi} (\pi_a q_a^\pi - \pi_a \sum_k \pi_k q_k^\pi) \\ &= \frac{\pi_a}{v^\pi} (q^{\pi_a} - v^\pi) \end{aligned} \tag{13}$$

which is the Maynard-Smith Replicator Dynamic. Similar to Proposition 1, we can further show that this update is the expected update performed by the 1-MCL algorithm, that we simply call *Maynard-Cross Learning* (MCL) for conciseness:

$$\begin{aligned}
\mathbb{E}[d^{\text{MCL}}\pi_a] &= \sum_{k=1}^{n} \pi_k . d^{\text{MCL}}\pi_a(k) \\
&= \pi_a . d^{\text{MCL}}\pi_a(a) + \sum_{k \neq a} \pi_k . d^{\text{MCL}}\pi_a(k) \\
&= \pi_a \frac{\mathbb{E}[r_a]}{v^\pi}(1 - \pi_a) + \sum_{k \neq a} \pi_k \frac{\mathbb{E}[r_k]}{v^\pi}(-\pi_a) \\
&= \frac{\pi_a}{v^\pi}\Big[\mathbb{E}[r_a] - \pi_a \mathbb{E}[r_a] - \sum_{k \neq a} \pi_k \mathbb{E}[r_k]\Big] \\
&= \frac{\pi_a}{v^\pi}\Big[\mathbb{E}[r_a] - \sum_{k} \pi_k \mathbb{E}[r_k]\Big] \\
&= \frac{\pi_a}{v^\pi}(q_a^\pi - v^\pi) \qquad (14)
\end{aligned}$$

Thus we can write the RL update of the "hive mind" in a swarm of bees following the R_{wvoter} model:

Proposition 2. An infinite population of individuals following R_{wvoter} can equivalently be seen as an RL agent following Exact Maynard-Cross Learning, i.e.,

$$d^{\text{wvoter}}\pi_a = \mathbb{E}[d^{\text{MCL}}\pi_a], \qquad (15)$$

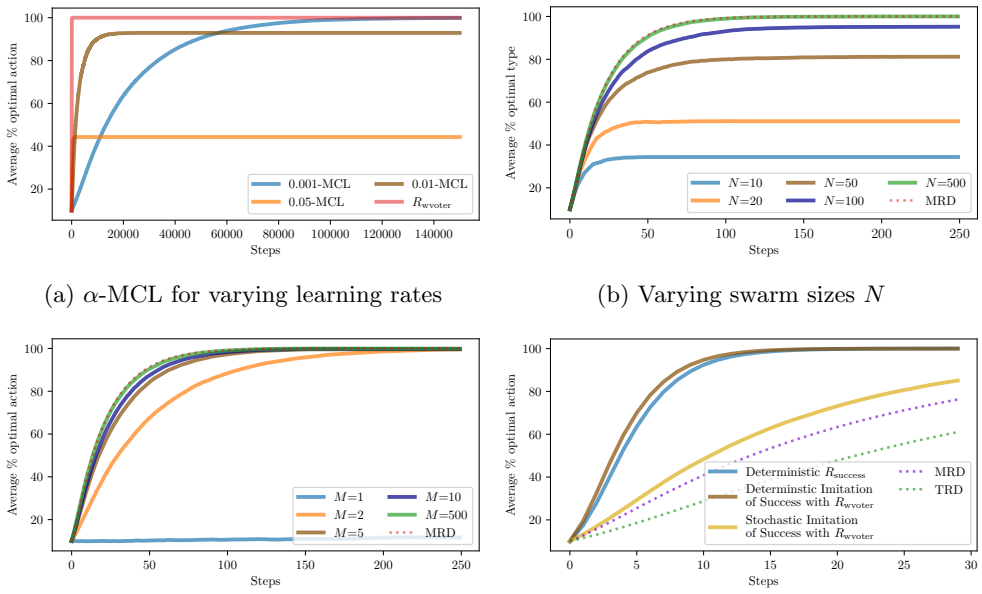
where π is both a population vector and a vector of action-probabilities, $d^{\text{wvoter}}\pi$ is the single-step change in the population vector π under the weighted voter revision model, and $d^{\text{MCL}}\pi$ is the update performed by MCL on the policy π.

Proof Direct consequence of Eqs. (13) and (14). □

4 Final remarks and simulations

4.1 A single RL agent in many parallel environments

Researchers have already informally witnessed collective intelligence [35, 36] described in Section 3, often referred to as "emergent" [37] to describe situations where simple, seemingly blind individual behaviors aggregate to a complex and coherent behavior of the group. Proposition 2 shows that, when the individual bees' behavior is modeled as R_{wvoter}, this "emergent" collective intelligence is MCL, a multi-armed bandit Reinforcement Learning algorithm. But the implications of Proposition 2 go even further: not only a swarm of bees is a single MCL agent, it is one that is quite efficient at what it does, because each bee is a parallel action sample of its policy. In other words,



(a) α-MCL for varying learning rates

(b) Varying swarm sizes N

(c) Varying neighborhood sizes for N=500

(d) Variants of R_{success} & R_{wvoter} in N=1000

Fig. 2: **Simulations**: (a) Swarms of bees reach consensus more rapidly when following R_{wvoter} collectively than when individual bees learn via α-MCL. (b,c) Varying swarm and neighborhood sizes show that the theoretical predictions hold under practical constraints. (d) Simple variants of R_{success} and R_{wvoter} can surpass R_{wvoter} in this environment, raising the question of why evolution favored R_{wvoter} over alternative decision-making strategies.

each bee i can be seen as an action sample $a^{(i)} \sim \pi$ tested against a parallel copy $E^{(i)}$ of the environment E, as illustrated in Fig. 1. Fig. 2a shows that this converges much faster than if individual bees were themselves to learn via α-MCL instead of just following R_{wvoter}. If the MCL algorithm were of any use outside of biology[2], say if a computer scientist wanted to train a robot via iterative α-MCL, they would need a small α for the algorithm to converge to the optimal option. Or, if feasible, they could afford using a larger α by evaluating the policy in several parallel simulations (see Sections D.5 and D.6). Swarms of honey bees naturally learn via parallel RL in as many environments as there are scouts in the swarm. Finally, Proposition 1 implies that a similar discussion holds for the "imitation of success" revision protocol, which encompasses certain human behaviors, and even natural selection itself.

4.2 Swarm size

In Section 4.1, we have discussed how being made of a number of individuals enables the "hive mind" to evaluate actions sampled from π in a parallel fashion. Furthermore,

[2]Most likely not: more advanced algorithms exist to solve multi-armed bandits (such as UCB [9]).

our analysis assumed that this number was large (which enabled us to compute flows in Section 3). Yet, in practice there is a tradeoff here. Actual honey bee swarms are known to use only a small proportion of their bees as scouts, while the large majority of the swarm remains quiescent during nest-hunting [38, 39], which is surmised to be in the interest of energy-efficiency. In a typical swarm of 10,000 bees, roughly 200 to 500 scouts are active during nest-site selection. To analyze how the number of scouts N affects the task-performance of the MCL macro-agent, we conducted simplified simulations where swarms of varying sizes had to choose the best amongst 10 options. Option qualities were spread between 0 and 1, while the noise in bees' estimates was modeled with a uniform perturbation of amplitude 0.2. Fig. 2b reports the percentage of populations that converged to the optimal choice for different values of N, across 1000 seeds (each individual seed converged to a homogeneous choice). In these simulations, $N = 500$ is enough to closely follow the MRD and converge to the optimal option, whereas a number of scout bees that is too small often yields convergence to a sub-optimal nest site option. However, our choice of simulation parameters is largely arbitrary: Fig. 2b should be interpreted qualitatively rather than quantitatively.

4.3 Neighborhood size

In Section 3, we have assumed that the local neighborhood size M was large. This assumption was useful to analytically derive the macro-agent RL update, because it meant that local neighborhoods (i.e., other scout bees that a scout may randomly walk into within its vicinity) could be considered well-mixed. Arguably, actual neighborhood sizes are not that large in the real world. Therefore, to complete our theoretical analysis, we performed R_{wvoter} simulations with varying neighborhood sizes. Fig. 2c shows that even a small $M \geq 5$ yields a macro-agent algorithm that closely follows MCL. With $M = 1$, each scout copies an arbitrary neighbor, which yields no macro-dynamic.

4.4 The collective power of promoting

The R_{wvoter} revision protocol adopted by honey bees has surprising advantages over imitation of success. In Section D, we show that R_{wvoter} (MCL) generally converges faster than R_{success} (CL), except for small population sizes. This is in addition to the simplicity of R_{wvoter}, which only requires bees to blindly mimic their peers.

4.5 Could bees do more?

In Section 3, we showed that the bees employing R_{wvoter} can be seen as a single online RL agent. However, a natural question arises: is R_{wvoter} the optimal collective-decision making strategy? To answer this question, we investigate three simple variants of the imitation of success model and the weighted voter model (more details in Section D.3):

- Deterministic Imitation of Success: Unlike the standard imitation of success model (which switches stochastically based on partner's success), this variant deterministically adopts the partner's type when it has more success.

- Deterministic Imitation of Success with Weighted Voter Rule: This variant combines the weighted voter model's success-weighted neighbor sampling with the deterministic imitation of success switch based on comparative success.
- Stochastic Imitation of Success with Weighted Voter Rule: This variant differs from Deterministic Imitation of Success with Weighted Voter Rule in that it uses classic stochastic imitation of success instead of our deterministic version as the switching rule.

Fig. 2d shows that, at least in this simplified environment, all these variants converge to the optimal decision faster than R_{wvoter} or MRD. This simple experiment suggests that there may be better collective decision-making strategies than R_{wvoter}. In particular, we see that if the bees could perform comparisons with their individual quality estimates before blindly imitating others from the neighborhood, the convergence speed could be faster than MRD. This insight raises an important question: why did bees evolve to use R_{wvoter} [19, 23] over other collective decision-making strategies?

We speculate that this behaviour relates to cognitive constraints and the fundamental challenge of quality comparison. The weighted voter model requires remarkably minimal individual capabilities: bees need only estimate the quality of their own option and observe the types (not quality estimates) of neighbors, broadcasting their preference at a frequency proportional to quality. R_{wvoter} requires no comparison operations between quality values and no direct communication of numerical quality estimates, capabilities that all faster-converging variants demand. In contrast, deterministic switching rules require bees to not only observe others' quality estimates and compare them to their own, but also presume an objective or at least mutually calibrated scale for evaluating nest site quality. This is non-trivial: individual bees may experience different environmental conditions during their assessments, have different sensory acuity, or weight various quality features differently. R_{wvoter} may elegantly sidestep this calibration problem by converting subjective quality estimates into broadcast frequencies — a transformation that preserves relative preferences while eliminating the need for interpersonal comparison. The weighted voter model may thus represent an evolutionary solution that achieves collective reinforcement learning with the absolute minimum individual cognitive machinery while avoiding the difficult problem of quality scale calibration, trading some convergence speed for dramatic simplification of individual processing and robustness to individual variation.

This hypothesis aligns with the broader pattern in social insects where complex collective behaviors emerge from remarkably simple individual rules, suggesting that evolutionary pressures favor cognitive parsimony at the individual level when collective intelligence can compensate at the group level, a strategy that may be driven in part by the substantial metabolic costs of neural tissue, which in mammals requires nearly an order of magnitude more energy per unit weight than other somatic tissues [40, 41]. We leave this exploration open for future work.

4.6 Broader significance

- **Biology.** Proposition 2 shows that seemingly blind, imitation-based bee interactions aggregate to a form of collective intelligence, which is that of a "hive mind"

learning through reinforcement. This is interesting from a neuroscience perspective, since both imitation and reinforcement have long been known to govern learning at the individual level [4, 42]. What our paper shows about honey bees is that their puzzling imitative behavior can in fact be understood as reinforcement from the swarm-agent perspective, where this behavior equates to massively parallel MCL. Why this imitative behavior has evolved in nature may thus appear clearer: swarms of honey bees are an example of reinforcement strategy having evolved at the group level. This analysis agrees with the commonly accepted theory of group/kin selection to explain the behavior of honey bees [43]. Building on the framework established in Proposition 1, parallel findings in other phylogenetically distant biological species underscore the generality of the proposed population–policy equivalences of this work. Notably, pheromone-guided C. elegans foraging exhibits the same mathematical structure as cross-learning [44], suggesting that stigmergic coordination may also be modeled as a form of collective reinforcement learning.

- **Economy and Social Dynamics.** Evolutionary Game Theory goes beyond modeling biology: it extends to many types of population dynamics, most notably in economics [29, 33]. When thinking of R_{success} as a model of certain decision-making strategies, Proposition 1 yields that imitating others' success makes us part of a macroscopic RL process that hugely benefits the whole group, by mutualizing our speed of convergence to optimal strategies (see Section 4.1). This opens an avenue for modeling the group-level impacts of economic actors' imitative behaviors as forms of collective reinforcement learning.
- **Algorithmic Swarm Intelligence.** The field of Swarm Intelligence takes inspiration from complex emergent behaviors arising from a collective of natural entities following simple, local, and decentralized rules [35], to engineer algorithms that leverage collective principles like coordination, cooperation, and communication to tackle various problems (including CDM) across multiple domains such as swarm robotics [45, 46] and optimization [47]. Propositions 1 and 2 provide theoretical grounding for the phenomenon commonly referred to as "emergence" in this field [37]. This bridge also opens up the possibility of porting ideas from Reinforcement Learning to derive Swarm Intelligence local rules.
- **Algorithmic Reinforcement Learning.** Similar to how we abstracted a population of R_{wvoter} individuals as a single MCL agent, other revision models (majority rule [19] and cross-inhibition [48]) can be used to abstract dynamic groups of entities as simple, single RL agents. In massively multi-agent real-world settings (e.g., finance), this may greatly help model the non-stationarity of environment dynamics.

4.7 Conclusion

We have demonstrated that imitation-based collective behavior in large populations can be mathematically equivalent to reinforcement learning, providing a unifying framework that reinterprets swarm intelligence, social dynamics, and evolutionary processes as emergent forms of collective learning. This result provides a formal basis for understanding how simple, local imitation behaviors, often regarded as non-cognitive, can give rise to group-level intelligence and adaptive learning at scale. In particular, Proposition 2 demonstrates that the "hive mind" observed in swarms of bees can

indeed be viewed as a single coherent agent learning via reinforcement, giving rise to a new bandit algorithm that we coin *Maynard-Cross Learning*.

References

[1] Neumann, J.V., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press, Princeton, NJ, USA (1944)

[2] Silver, D., Singh, S., Precup, D., Sutton, R.S.: Reward is enough. Artificial Intelligence **299**, 103535 (2021) https://doi.org/10.1016/j.artint.2021.103535

[3] Sutton, R.S., Barto, A.G.: Reinforcemet Learning: An Introduction. A Bradford Book, Cambridge, MA, USA (2018)

[4] Neftci, E.O., Averbeck, B.B.: Reinforcement learning in artificial and biological systems. Nature Machine Intelligence **1**(3), 133–143 (2019)

[5] Muller, T.H., Butler, J.L., Veselic, S., Miranda, B., Wallis, J.D., Dayan, P., Behrens, T.E.J., Kurth-Nelson, Z., Kennerley, S.W.: nature neuroscience distributional reinforcement learning in prefrontal cortex. Nature Neuroscience — **27**, 403–408 (2024) https://doi.org/10.1038/s41593-023-01535-w

[6] Elsayed, M., Vasan, G., Mahmood, A.R.: Streaming Deep Reinforcement Learning Finally Works (2024). https://arxiv.org/abs/2410.14606

[7] Vasan, G., Elsayed, M., Azimi, A., He, J., Shariar, F., Bellinger, C., White, M., Mahmood, A.R.: Deep Policy Gradient Methods Without Batch Updates, Target Networks, or Replay Buffers (2024). https://arxiv.org/abs/2411.15370

[8] Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning, pp. 1928–1937 (2016). PmLR

[9] Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Machine Learning **47**, 235–256 (2002) https://doi.org/10.1023/A:1013689704352

[10] Williams, R.J.: Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning (1992)

[11] Cross, J.G.: A stochastic learning model of economic behavior. The Quarterly Journal of Economics **87**(2), 239–266 (1973)

[12] Bloembergen, D., Tuyls, K., Hennes, D., Kaisers, M.: Evolutionary dynamics of multi-agent learning: A survey. Journal of Artificial Intelligence Research **53**, 659–697 (2015) https://doi.org/10.1613/jair.4818

[13] Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. **8**(3–4), 229–256 (1992) https://doi.org/10.1007/BF00992696

[14] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal Policy Optimization Algorithms (2017). https://arxiv.org/abs/1707.06347

[15] Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor (2018). https://arxiv.org/abs/1801.01290

[16] Reina, A., Valentini, G., Fernández-Oto, C., Dorigo, M., Trianni, V.: A design pattern for decentralised decision making. PLOS ONE **10**(10), 1–18 (2015) https://doi.org/10.1371/journal.pone.0140950

[17] Bose, T., Reina, A., Marshall, J.A.: Collective decision-making. Current Opinion in Behavioral Sciences **16**, 30–34 (2017) https://doi.org/10.1016/j.cobeha.2017.03.004 . Comparative cognition

[18] Jackson, M., Golub, B.: Naïve learning in social networks and the wisdom of crowds. American Economic Journal: Microeconomics **2**, 112–49 (2010) https://doi.org/10.1257/mic.2.1.112

[19] Valentini, G., Ferrante, E., Hamann, H., Dorigo, M.: Collective decision with 100 Kilobots: speed versus accuracy in binary discrimination problems. Autonomous Agents and Multi-Agent Systems **30**(3), 553–580 (2016) https://doi.org/10.1007/s10458-015-9323-3

[20] Valentini, G., Hamann, H., Dorigo, M.: Self-organized collective decision making: the weighted voter model. In: Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems. AAMAS '14, pp. 45–52. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2014)

[21] Sandholm, W.H.: Population Games and Evolutionary Dynamics. The MIT Press, Cambridge, MA (2010). http://www.jstor.org/stable/j.ctt5hhbq5 Accessed 2025-05-23

[22] Reina, A., Njougouo, T., Tuci, E., Carletti, T.: Speed-accuracy trade-offs in best-of-n collective decision making through heterogeneous mean-field modeling. Phys. Rev. E **109**, 054307 (2024) https://doi.org/10.1103/PhysRevE.109.054307

[23] Visscher, P.K., Camazine, S.: Collective decisions and cognition in bees. Nature **397**(6718), 400 (1999) https://doi.org/10.1038/17047

[24] Seeley, T.D., Visscher, P.K.: Quorum sensing during nest-site selection by honeybee swarms. Behavioral Ecology and Sociobiology **56**, 594–601 (2004)

[25] Seeley, T.D., Visscher, P.K.: Group decision making in nest-site selection by honey bees. Apidologie **35**(2), 101–116 (2004)

[26] Passino, K.M., Seeley, T.D.: Modeling and analysis of nest-site selection by honeybee swarms: the speed and accuracy trade-off. Behavioral Ecology and Sociobiology **59**, 427–442 (2006)

[27] Passino, K.M., Seeley, T.D., Visscher, P.K.: Swarm cognition in honey bees. Behavioral Ecology and Sociobiology **62**(3), 401–414 (2008). Accessed 2025-05-04

[28] Sandholm, W.H., Dokumacı, E., Lahkar, R.: The projection dynamic and the replicator dynamic. Games and Economic Behavior **64**(2), 666–683 (2008) https://doi.org/10.1016/j.geb.2008.02.003 . Special Issue in Honor of Michael B. Maschler

[29] Apesteguia, J., Huck, S., Oechssler, J.: Imitation—theory and experimental evidence. Journal of Economic Theory **136**(1), 217–235 (2007)

[30] Taylor, P.D., Jonker, L.B.: Evolutionary stable strategies and game dynamics. Mathematical Biosciences **40**(1), 145–156 (1978) https://doi.org/10.1016/0025-5564(78)90077-9

[31] Nowak, M.A.: Five rules for the evolution of cooperation. Science **314**(5805), 1560–1563 (2006) https://doi.org/10.1126/science.1133755 https://www.science.org/doi/pdf/10.1126/science.1133755

[32] Smith, J.M.: Evolution and the Theory of Games. Cambridge University Press, Cambridge, UK (1982)

[33] Börgers, T., Sarin, R.: Learning through reinforcement and replicator dynamics. Journal of economic theory **77**(1), 1–14 (1997)

[34] Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., State, G.: Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning (2021). https://arxiv.org/abs/2108.10470

[35] Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, New York (1999). https://doi.org/10.1093/oso/9780195131581.001.0001 . https://doi.org/10.1093/oso/9780195131581.001.0001

[36] Franks, N.R., Pratt, S.C., Mallon, E.B., Britton, N.F., Sumpter, D.J.: Information flow, opinion polling and collective intelligence in house–hunting social insects. Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences **357**(1427), 1567–1583 (2002)

[37] De Wolf, T., Holvoet, T.: Emergence and self-organisation: a statement of similarities and differences. In: Proceedings of the International Workshop on Engineering Self-organising Applications 2004, pp. 96–110 (2004)

[38] Seeley, T.D., Morse, R.A.: Nest site selection by the honey bee, apis mellifera. Insectes Sociaux **25**(4), 323–337 (1978)

[39] Beekman, M., Fathke, R.L., Seeley, T.D.: How does an informed minority of scouts guide a honeybee swarm as it flies to its new home? Animal behaviour **71**(1), 161–171 (2006)

[40] Rittschof, C.C., Schirmeier, S.: Insect models of central nervous system energy metabolism and its links to behavior. Glia **66**(6), 1160–1175 (2018)

[41] Schippers, M.-P., Dukas, R., Smith, R., Wang, J., Smolen, K., McClelland, G.: Lifetime performance in foraging honeybees: behaviour and physiology. Journal of experimental biology **209**(19), 3828–3836 (2006)

[42] Zentall, T.R.: Imitation: definitions, evidence, and mechanisms. Anim Cogn **9**, 335–353 (2006) https://doi.org/10.1007/s10071-006-0039-2

[43] Jr, R.E., Robinson, G., Fondrk, M.: Genetic specialists, kin recognition and nepotism in honey-bee colonies. Nature **338**, 576–579 (1989) https://doi.org/10.1038/338576a0

[44] Vellinger, A., Antonic, N., Tuci, E.: From Pheromones to Policies: Reinforcement Learning for Engineered Biological Swarms (2025). https://arxiv.org/abs/2509.20095

[45] Dorigo, M., Theraulaz, G., Trianni, V.: Swarm robotics: Past, present, and future [point of view]. Proceedings of the IEEE **109**(7), 1152–1165 (2021) https://doi.org/10.1109/JPROC.2021.3072740

[46] Hamann, H.: Swarm Robotics: A Formal Approach, 1st edn. Springer, Cham (2018)

[47] Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. IEEE Computational Intelligence Magazine **1**(4), 28–39 (2006) https://doi.org/10.1109/MCI.2006.329691

[48] Reina, A., Marshall, J.A.R., Trianni, V., Bose, T.: Model of the best-of-n nest-site selection process in honeybees. Physical Review E **95**(5) (2017) https://doi.org/10.1103/physreve.95.052411

[49] Soma, K., Vardharajan, V.S., Hamann, H., Beltrame, G.: Congestion and scalability in robot swarms: A study on collective decision making. In: 2023 International Symposium on Multi-Robot and Multi-Agent Systems (MRS), pp. 199–206

(2023). https://doi.org/10.1109/MRS60187.2023.10416793

[50] Prasetyo, J., Masi, G.D., Ferrante, E.: Collective decision making in dynamic environments. Swarm Intelligence **13**, 217–243 (2019) https://doi.org/10.1007/s11721-019-00169-8

[51] Ebert, J.T., Gauci, M., Nagpal, R.: Multi-feature collective decision making in robot swarms. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. AAMAS '18, pp. 1711–1719. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2018)

[52] Raoufi, M., Hamann, H., Romanczuk, P.: Speed-vs-accuracy tradeoff in collective estimation: An adaptive exploration-exploitation case. In: 2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS), pp. 47–55 (2021). https://doi.org/10.1109/MRS50823.2021.9620695

[53] Ebert, J., Gauci, M., Mallmann-Trenn, F., Nagpal, R.: Bayes bots: Collective bayesian decision-making in decentralized robot swarms. In: 2020 IEEE International Conference on Robotics and Automation, ICRA 2020. Proceedings - IEEE International Conference on Robotics and Automation, pp. 7186–7192. Institute of Electrical and Electronics Engineers Inc., United States (2020). https://doi.org/10.1109/ICRA40945.2020.9196584 . Publisher Copyright: © 2020 IEEE.; 2020 IEEE International Conference on Robotics and Automation, ICRA 2020 ; Conference date: 31-05-2020 Through 31-08-2020

[54] Pirrone, A., Reina, A., Stafford, T., Marshall, J.A.R., Gobet, F.: Magnitude-sensitivity: rethinking decision-making cognitive sciences. Trends in Cognitive Sciences **26**, 66–80 (2022) https://doi.org/10.1016/j.tics.2021.10.006

[55] Coucke, N., Heinrich, M.K., Cleeremans, A., Dorigo, M., Dumas, G.: Collective decision making by embodied neural agents. PNAS Nexus **4**(4), 101 (2025) https://doi.org/10.1093/pnasnexus/pgaf101

A Proofs for Section 3 (Methodology)

Lemma 1. An infinite population of individuals adopting R_{success} follows the TRD:

$$d\pi_a = \pi_a(q_a^\pi - v^\pi), \tag{5}$$

where π_a is the proportion of type a in the population, q_a^π is the expected payoff (also called "fitness") of type a against the population, and v^π is the average fitness of the population.

Proof. Let $P(a \leftarrow b)$ denote the inflow of individuals of type b into type a, i.e, the proportion of the population leaving type b and adopting type a. The population has a proportion of π_b individuals of type b, each having a probability π_a of meeting an individual of type a, and a conditional probability $\mathbb{E}[r_a]$ of switching to its type. Thus, we get $P(a \leftarrow b) = \pi_b \pi_a \mathbb{E}[r_a]$:

$$\begin{aligned}
d\pi_a &= \sum_{b \neq a} \underbrace{P(a \leftarrow b)}_{\text{inflow}} - \underbrace{P(b \leftarrow a)}_{\text{outflow}} && (16)\\
&= \sum_{b \neq a} \pi_b \pi_a \mathbb{E}[r_a] - \pi_a \pi_b \mathbb{E}[r_b] \\
&= \pi_a \Big[\sum_{b \neq a} \pi_b \mathbb{E}[r_a] - \sum_{b \neq a} \pi_b \mathbb{E}[r_b] \Big] \qquad \sum_{b \neq a} \pi_b + \pi_a = 1 \\
&= \pi_a \Big[(1 - \pi_a)\mathbb{E}[r_a] - \sum_{b \neq a} \pi_b \mathbb{E}[r_b] \Big] \\
&= \pi_a \Big[\mathbb{E}[r_a] - \sum_{b} \pi_b \mathbb{E}[r_b] \Big] \\
&= \pi_a (q_a^\pi - v^\pi) && (17)
\end{aligned}$$

□

Lemma 2. In expectation, an RL agent learning via the CL update rule follows:

$$\mathbb{E}[d\pi_a] = \pi_a(q_a^\pi - v^\pi), \tag{6}$$

where q_a^π is the action-value of a, and v^π is the value of policy π.

Proof. Let us compute the expectation over actions sampled from π in Eq. 2. For convenience, we write
$\mathbb{E}[d\pi_a] := \mathbb{E}_{k \sim \pi}[d\pi_a(k)]$, and $\mathbb{E}[r_k] := \mathbb{E}_{r_k \sim r(k)}[r_k]$:

$$\mathbb{E}[d\pi_a] = \sum_{k=1}^{n} \pi_k . d\pi_a(k) \tag{18}$$

$$
\begin{aligned}
&= \pi_a.d\pi_a(a) + \sum_{k \neq a} \pi_k.d\pi_a(k) \\
&= \pi_a \mathbb{E}[r_a](1 - \pi_a) + \sum_{k \neq a} \pi_k \mathbb{E}[r_k](-\pi_a) \\
&= \pi_a \Big[\mathbb{E}[r_a] - \pi_a \mathbb{E}[r_a] - \sum_{k \neq a} \pi_k \mathbb{E}[r_k] \Big] \\
&= \pi_a \Big[\mathbb{E}[r_a] - \sum_{k} \pi_k \mathbb{E}[r_k] \Big] \\
&= \pi_a (q_a^\pi - v^\pi) \qquad (19)
\end{aligned}
$$

□

B Collective Decision-Making in Swarm Robotics

This section briefly explores the connections between Collective Decision-Making and Swarm Intelligence. This will help realize the extent of the impact that the established bridge can have on the field of swarm robotics. Drawing inspiration from the nest-site selection behavior of honey bees, swarm intelligence researchers have created collective decision-making strategies for robot swarms. Such collective decision-making systems have applications such as monitoring of forest fires, patrolling oceans, etc. A representative setup [49] consists of two spatially separated zones and a central nest zone. The quality of each sampling zone is represented by certain features within the sampling zone (such as the color of the zone, etc.). Robot swarms are then tasked with identifying the highest quality zone by combining exploration, sampling, and communication behaviors. Simple robots are typically used that execute simple behaviors, such as phototaxis (motion towards or away from a source of light), to commute between the zones. Sampling behaviors are executed in the zones using simple sensors, which give noisy quality estimates of the zone. These estimates enable the robot to form opinions of the sampled zone. After sampling, robots return to the nest and perform random movements, facilitating population mixing and opinion diffusion. To simulate the "waggle dance", robots modulate the amount of time they spend in the nest, whereby they spend time proportional to the quality of the zone (referred to as positive modulation). Robots can then use different protocols, such as the weighted voter model, to update their individual opinions. Upon repeated such interactions, the robot swarm converges to the optimal decision. This setting also forms an excellent testbed for examining the influence of finite size effects, congestion, and connectivity constraints of the swarm. Apart from the weighted voter model described in this work, there exist other decision-making strategies, such as the majority rule model [19], where individuals collect opinions of their immediate local neighbors and switch to the majority opinion. Cross-inhibition [48] is another model that involves various mechanisms such as recruitment (similar to positive modulation) and inhibition (stop signaling to inhibit individuals from switching their opinions) to make decisions. Beyond these foundational work, investigations related to dynamic qualities for options [50], multi-feature qualities for options [51], continuous space options [52], Bayesian approaches to model

beliefs of individuals for options [53], and quality magnitude sensitivity based studies [54] have been carried out in the literature. In addition, investigations have also been carried out that model simple neural dynamics for sensorimotor coordination of agents [55] for collective decision-making.

C Notations

Since this paper uses nomenclature from different fields to denote similar things, we summarize them in one place for quick reference. This is similar to the table used by Bloembergen et al [12].

Reinforcement learning	Collective-Decision Making/Evolutionary Game Theory
action	opinion/option/type
policy	population vector
reward	quality/payoff
RL agent	hive-mind

Table 1: Overlapping nomenclature between the fields of RL, CDM, and EGT.

D Simulations

In this section, we provide some additional results to consolidate the theory presented in the manuscript. To do so, we first describe the two RL update rules: CL and MCL, in both their streaming and parallel variants. We also outline the implementation details of the population update rules R_{success} and R_{wvoter} along with their variants. Finally, we numerically simulate the TRD and MRD to compare the RL and population update rules against their corresponding analytical solutions. It is important to note that MCL is not intended as a competitive bandit algorithm, but only as a biologically motivated construct designed to demonstrate that an imitation-based population update (R_{wvoter}) can be captured by an RL update rule.

D.1 Environment

We consider the standard multi-armed stateless bandit setting described in preliminaries (see Section 2.1). It is clear from the Population-policy equivalence remark (Remark 1) that we can use the same environment for RL and population experiments. The environment returns a noisy reward signal sampled from the hidden distribution $r(a)$ when action a is taken. We define the hidden reward distribution as a uniform distribution, given by $r \sim \mathcal{U}(q_a^\pi - \Delta, q_a^\pi + \Delta)$, where q_a^π is the mean reward associated with action a, and Δ controls the amplitude of noise around q_a^π. To ensure the validity of both RL and population updates, rewards are bounded within the interval [0,1], which imposes the constraint $q_a^\pi - \Delta \geq 0$ and $q_a^\pi + \Delta \leq 1$. We consider three distinct scenarios for the reward means across actions: (i) Low, where all q_a^π values are

equally spaced in the range [0.1, 0.4], (ii) Middle, where q_a^π values are equally spaced in the range [0.4,0.7], and (iii) High, where all q_a^π values are equally spaced in the range [0.6,0.9].

D.2 RL Experiments

Streaming: In these experiments, an RL agent interacts with a single environment in a streaming fashion. The RL agent starts with an initial random policy π. The agent then samples one action k at each computation step from π in an iterative fashion. For pulling this action k, the agent receives a noisy reward signal $r_k \sim r(k)$ from the environment. Subsequently, for CL, the agent utilizes Eq. (1) with learning rate α to update the policy.

$$\forall a, \pi_a \leftarrow \pi_a + \alpha r_k \begin{cases} 1 - \pi_a & \text{if } a = k \\ -\pi_a & \text{otherwise} \end{cases} \tag{20}$$

Whereas for MCL, Eq. (12) cannot be used directly, since v^π is not estimated. Therefore, v^π is approximated by employing a moving average over rewards, where γ is a weighting factor for recent rewards:

$$\bar{r} \leftarrow \gamma r + (1 - \gamma)\bar{r} \; . \tag{21}$$

Moreover, since this update rule can make π invalid, i.e., components could become negative or above one, we clamp π between 0 and 1:

$$\forall a, \pi_a \leftarrow \text{clamp} \left(\pi_a + \alpha \frac{r_k}{\bar{r}} \begin{cases} 1 - \pi_a & \text{if } a = k \\ -\pi_a & \text{otherwise} \end{cases} \right) \tag{22}$$

Where α is the learning rate. These computations are carried out for every training *step*, and there are S steps per iteration.

Parallel: In these experiments, we implement parallel variants of the CL and MCL update rules, referred to as P-CL and P-MCL hereafter, respectively. P-CL performs a straightforward parallelization of the CL rule: at each update step, the policy π is updated based on the average effect of B independent samples collected from B parallel environments, as if simulating the CL update B times in parallel and averaging the results. Similarly, in P-MCL, the policy update is computed using the average effect of B parallel samples, normalized by v^π. Unlike streaming MCL, where normalization (v^π) is based on a moving average of past rewards, P-MCL uses the current mean of the B parallel rewards as the normalizing factor. With P-MCL, we also need to explicitly limit these policy updates between 0 and 1 to ensure that π remains valid. These calculations are also performed for S training steps per iteration.

D.3 Population Experiments

$R_{\textbf{success}}$: We implement R_{success} from Section 2.2.1. We start with an equal proportion of individuals associated with any type. Further, each individual receives a stochastic

payoff estimate ($r \sim r(a)$) for their type. Then, at each step, everyone is paired with another random individual for imitation. All individuals then generate a random real number between 0 and 1, and if the random real number is greater than the payoff of their paired individual, imitation is successful and they switch to their paired partner's type (rule 3 of R_{success}). If the generated random number is not greater than the payoff of their paired individual, they do not imitate and stick to their own types. Further, we consider another variant **Deterministic Imitation of Success** where individuals deterministically switch to the imitating partner's type if the rewards of the partner are higher than their rewards. There are S decision-making steps per iteration.

$R_{\textbf{wvoter}}$: We implement R_{wvoter} from Section 2.2.2. We start with an equal proportion of individuals associated with any opinion. Further, each individual receives a stochastic quality estimate ($r \sim r(a)$) for their opinion. Then at each step, each individual (i) switches to a opiton sampled from the distribution of votes cast $v^{(i)}$ (simulated effect of "waggle dance"), where $v_k^{(i)}$ is the ratio of votes cast for option k by the total number of votes cast in it's neighborhood ($\mathcal{N}^i$) excluding itself:

$$v_k^{(i)} = \frac{\sum\limits_{\forall p \in \mathcal{N}^i:\ T_k = i} r_p}{\sum\limits_{\forall q \in \mathcal{N}^i} r_q} \ . \tag{23}$$

These neighborhoods for each individual are formed by randomly sampling M individuals from $\mathcal{P}$ at every time step. It is to be noted that, unless specified, the neighborhood sizes are equal to the population size. Further, we consider two variants **Deterministic Imitation of Success with Weighted Voter rules** and **Stochastic Imitation of Success with Weighted Voter rules**, where we combine ideas from R_{wvoter} and R_{success}. With both of these variants, each individual is first paired with one of their neighbors with a probability proportional to the neighbor's rewards. After this pairing, with Stochastic Imitation of Success with Weighted Voter Rules, individuals switch similar to R_{success} and with Deterministic Imitation of Success with Weighted Voter rules, individuals switch similar to Deterministic Imitation of Success. There are S decision-making steps per iteration. The code for reproducing the simulations can be found here https://github.com/MISTLab/HiveMindRL.git

D.4 TRD and MRD

To empirically validate Propositions 1 and 2, we numerically simulate both the variants of RD according to Eqs. (3) and (4). As these equations are continuous, we discretize them by a step δ (discretizing step). Further, we start from an initial random population/policy (π) and simulate its evolution according to TRD and MRD between time intervals $[0, t_f]$, using the privileged information q_a^π not available to RL and population experiments.

$$\pi_a \leftarrow \pi_a + \delta \pi_a [q_a^\pi - \sum_l \pi_l q_l^\pi] \tag{24}$$

$$\pi_a \leftarrow \pi_a + \delta \frac{\pi_a}{v^\pi}[q_a^\pi - \sum_l \pi_l q_l^\pi] \tag{25}$$

Hyperparameter	value
Arms (n)	10
iterations	1000
Range of noise (2Δ)	0.2
Parallel Environments (B)	$\{10, 1000\}$
Discretizing factor (δ)	1

(a) Parallel RL experiments

Hyperparameter	value
Types/options (n)	10
Seeds	1000
Range of noise (2Δ)	0.2
Population size (N)	$\{10, 1000\}$
Discretizing factor (δ)	1

(b) Population experiments

Hyperparameter	value
Arms (n)	10
Learning rate (α)	$\{0.001, 0.1\}$
iterations	1000
Range of noise (2Δ)	0.2
Weight factor (γ)	0.01
Discretizing factor (δ)	α

(c) Streaming RL experiments

Hyperparameter	value
Arms (n)	10
Population size	1000
iterations	1000
Range of noise (2Δ)	0.2
Neighborhood sizes	$\{2, 10, 1000\}$
Discretizing factor (δ)	α

(d) Neighborhood experiments

Table 2: Hyperparameters for RL and population experiments.

In all the subsequent experiments, we track the evolution of optimal decisions in both reinforcement learning and population experiments. In the RL setting, we monitor the "% optimal action", defined as $\pi_a \times 100$ (where a is the optimal action), and report the average along with its standard deviation over all seeds at each step. Similarly, in the population setting, we compute the "% optimal type", defined analogously as $\pi_a \times 100$, where a is the optimal type. We also include the trajectories of the optimal action/type under the TRD and MRD dynamics to compare the empirical behavior of the RL and population update rules with their corresponding analytical models. The hyperparameters used in these simulations are provided in Table 2.

D.5 Streaming RL update rules follow analytical solutions when the learning rate is small

These results are presented in Fig. 3. For all scenarios, CL and MCL follow TRD and MRD, respectively, with small α, which can be explicitly seen with the dotted line of the analytical solutions (TRD, MRD) exactly at the center of the optimal % action curves of the CL and MCL update rules. This empirically validates that, with a small α, Eq. (1) with α and Eq. (12) follow the TRD and MRD, respectively, even in a streaming fashion. However, as soon as α increases, CL and MCL start deviating from their respective analytical solutions and have a huge standard deviation. This is a well-known effect in optimization literature. Interestingly, imagining the action samples forming a population (see Section 4.1), we see that a larger α corresponds to

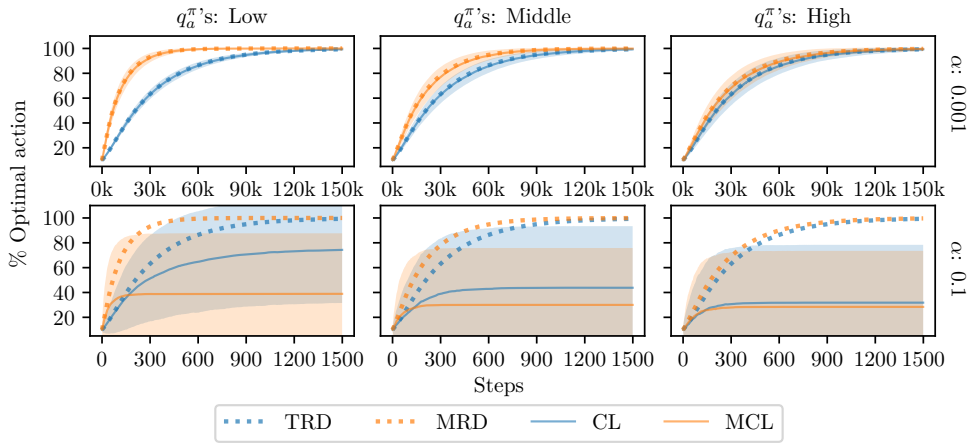


Fig. 3: Results for streaming RL experiments.

a smaller population, which leads to a poor approximation of the expected update.

D.6 Parallel RL update rules follow analytical solutions when the number of parallel environments is large.

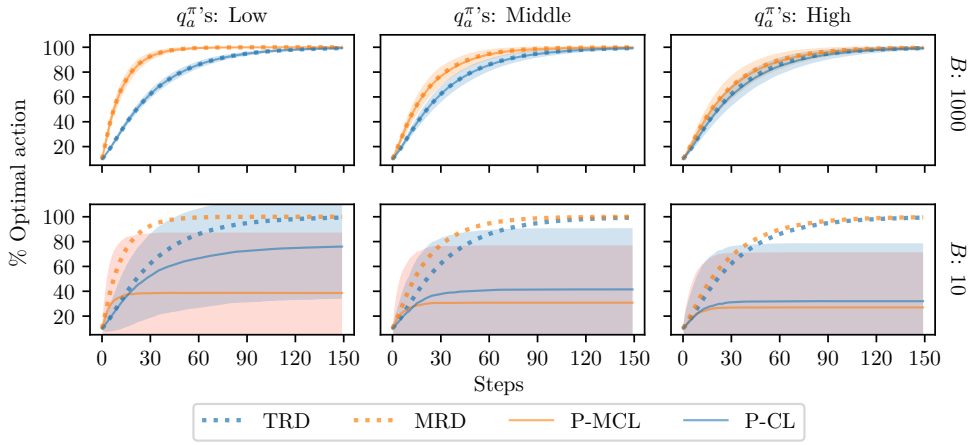


Fig. 4: Results for parallel RL experiments.

As seen in Fig. 4, it is clear that P-CL and P-MCL follow TRD and MRD, respectively, when updates are made utilizing a large number of parallel environments (this can be seen from the way the analytical solution is exactly at the center of the

% optimal action curves of P-CL and P-MCL). However, as soon as the number of parallel environments is reduced, the updates deviate from their analytical solutions (see Section 4.1).

D.7 Population update rules: R_{success} & R_{wvoter} follow their analytical solutions when the population sizes are large

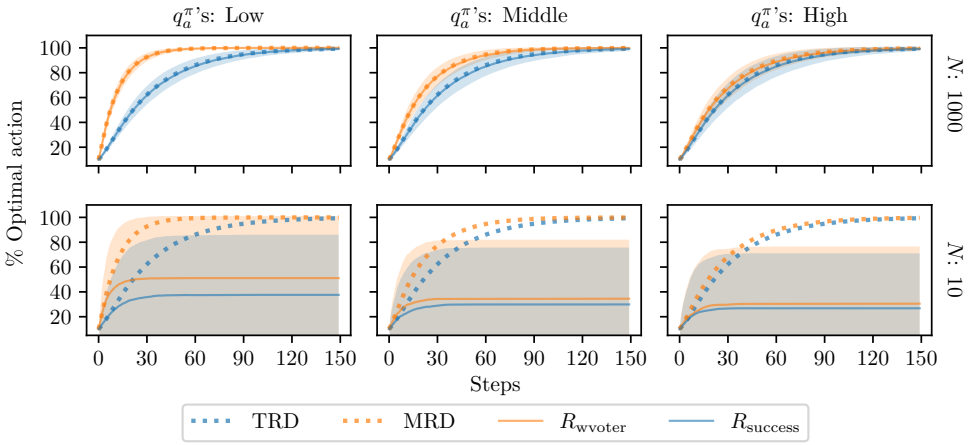


Fig. 5: Results for population experiments.

It can be seen in Fig. 5 that both R_{success} and R_{wvoter} follow TRD and MRD respectively when the population size is large. As soon as the population shrinks, R_{success} and R_{wvoter} begin to deviate from the analytical solution.

D.8 R_{success} & R_{wvoter} variants

It can be seen in Fig. 6 that the variants Deterministic Imitation of Success and Deterministic Imitation of Success with Weighted Voter rules perform better than MRD for all scenarios. However, Stochastic Imitation of Success with Weighted Voter Rules performs better than MRD in the Middle and High scenario, and performs worse than MRD in the low scenario. This could be because the probability of imitation might be low for the Low scenario as the scales of rewards are smaller.

D.9 Neighbourhoood sizes in R_{wvoter} only affect the convergence speed to MRD

It can be seen in Fig. 7 that for a large enough population R_{wvoter} follows MRD for any neighborhood size. However, it can be seen that the convergence speed is affected by smaller neighborhood sizes.

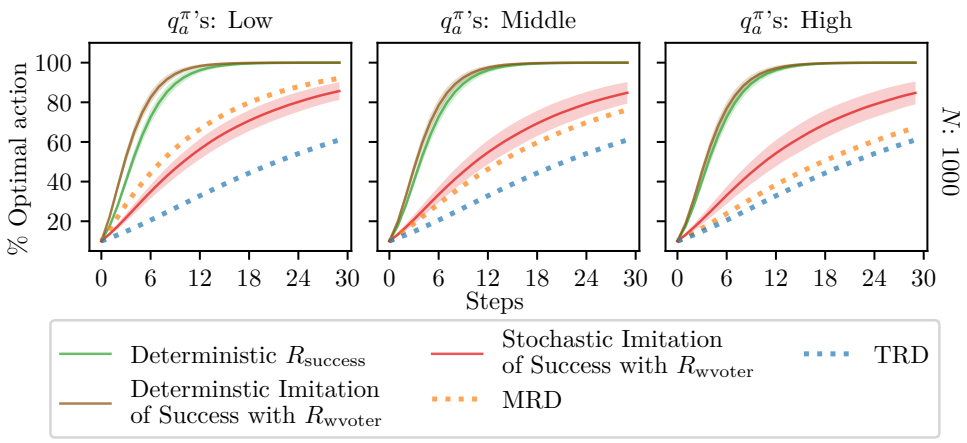


Fig. 6: Results for variants of population update rules for N=1000.

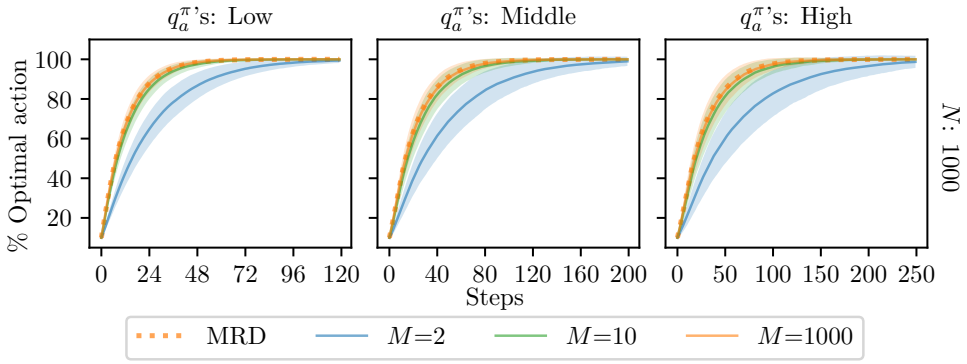


Fig. 7: Results for neighborhood experiments.

D.10 Convergence rate of MRD is $\geq$ TRD

As noted in [21], TRD and MRD can be rearranged in the form:

$$\dot{\pi_a} = v^\pi(\frac{\pi_a q_a^\pi}{v^\pi} - \pi_a) \qquad \text{(TRD)} \qquad (26)$$

$$\dot{\pi_a} = 1(\frac{\pi_a q_a^\pi}{v^\pi} - \pi_a) \qquad \text{(MRD)} \qquad (27)$$

$\dot{\pi}$ being the update "speed" and v^π being bounded between 0 and 1. The MRD speed is thus greater than the TRD speed for a given scenario. Empirically, we observe that MRD converges faster than TRD, especially when the q_a^π's are low and middle, as seen with any of the Figs. 3 to 5. Whereas, when the q_a^π's are high, there is very little difference (as $v^\pi \approx 1$). By extension, this also implies that MCL (for small α), P-MCL

(for large B), and R_{wvoter} (for large population) have convergence rates $\geq$ CL, P-CL, and R_{success}, respectively. However, to compare the convergence speeds of TRD and MRD across various reward scales, we revert the equations to their original form.

$$\dot{\pi_a} = \pi_a(q_a^\pi - v^\pi) \qquad \text{(TRD)} \tag{28}$$

$$\dot{\pi_a} = \pi_a\Big(\frac{q_a^\pi - v^\pi}{v^\pi}\Big) \qquad \text{(MRD)} \tag{29}$$

As the term $q_a^\pi - v^\pi$ denotes the relative fitness of any a (or advantage in the RL literature), we can see that TRD has a constant convergence speed across the reward scales. However, with MRD, this relative fitness is normalized by v^π, which increases with higher reward scales, leading to slower convergence speed with higher reward scales. This, combined with the observation that the speed of MRD $\geq$ TRD for any given scenario, shows that MRD catches up with TRD as the reward scales increase, as seen in Figs. 3 to 5.