

When the Hive Gets It Wrong: Where a Swarm of Imitators Stops Acting Like One Reinforcement Learner

Two reproductions, then experiments on small swarms, scouting and dishonest bees, set against earlier work on robot swarms.

Berg Atkinson, Wolfberg LLC · berg@wolfberg.ai

Preprint. Not peer reviewed. Published 30 September 2026.

IN PLAIN TERMS

When honey bees pick a new home, scouts visit candidate sites and come back and dance, longer and harder for a better site, and other bees copy the dances they run into. No bee compares sites. A 2024 paper by Soma, Bouteiller, Hamann and Beltrame proves that a very large swarm following this rule behaves, on average, exactly like a single reinforcement learning agent running a specific update rule they call Maynard-Cross Learning. On 30 September 2026, Ted Tanner walked through that result in a post with a working notebook and suggested the next experiment: make the choice harder and see what breaks.

We ran his notebook and every number reproduced exactly, and we rebuilt the paper's own swarm-size experiment from its public code and matched its published figure. Then we ran that experiment and several more, 35,166 simulated runs in all. The equivalence holds where the paper says it does, for a large swarm, and it bends in ways you can measure below that. A swarm that is too small for how close the options are will sometimes settle on the wrong one, and pure copying can never undo it. How small is too small depends on the gap between the options divided by the average payoff, not on the gap alone. A very small amount of random scouting prevents permanent wrong settlements. And because the swarm trusts every dance, a few bees that never change their minds can take it over, especially for an option almost as good as the best, even if they dance honestly. Checking a site yourself before you copy it is the defense against bees that lie, and it costs more the closer the options are.

Much of this was known. Earlier work on robot swarms had shown that bigger swarms of this kind choose better, and that a few stubborn or lying robots can steer one; Section 4 says what was known and what is new here.

ABSTRACT

Soma et al. (2024) show that an infinite population following the weighted voter model of honey bee nest-site selection moves, in expectation, like a single bandit learner using an

update they name Maynard-Cross Learning. We reproduced Tanner's (2026) notebook exactly and the source paper's swarm-size figure within sampling error, then ran 235 configurations (35,166 simulated runs) on five-armed bandits with uniform reward noise. A 1,000-bee swarm hearing every bee stays within 0.015 of the replicator curve. Smaller swarms settle on a worse site more often, as earlier work on this model found. The threshold is set by the number of bees times the quality gap divided by the average reward: shifting every reward moved it by the predicted factor, and with eight neighbors no run settled wrongly once that quantity reached 12.5. Dividing by the average reward only rescales the clock, which trades immunity to reward scale for sensitivity to reward shifts. Without exploration a wrong settlement is permanent; 0.2% random scouting removed it. A few bees that never switch decide close contests, and separating the two behaviors of the standard adversary shows that stubbornness, not exaggeration, does most of the damage: 2% stubborn but honest bees for the second-best site took 98% of the honest swarm. Checking a site by visiting it before switching defends against exaggerated reports, at a cost that grows as the options get closer. We relate the results to prior work on robot swarms and to how language models are trained with reinforcement learning.

Keywords: swarm intelligence, reinforcement learning, replicator dynamics, collective decision-making, Maynard-Cross Learning, robustness, adversarial agents

CONTENTS

1	Introduction	5	Methods
2	The claim and the math behind it	6	Results
3	How a reinforcement learning agent differs from the hive	7	What it means
4	What was already known	8	Limits
			References

1 Introduction

Soma, Bouteiller, Hamann and Beltrame (2024) write down what each bee does in a model of how honey bees choose a nest, add it up, and show that the sum is a learning rule. Their central result is that a large population of bees choosing a nest site by the weighted voter model moves, on average, exactly the way a single reinforcement learning agent's policy moves under an update they call Maynard-Cross Learning. In expectation, the swarm is one learner.

Ted Tanner (2026) took the paper apart in a post with a notebook anyone can run, and was careful where most treatments are not. He separated the claim that the swarm and the single learner reach the

same destination from the claim that they get there on the same schedule, and he warned against reading the evolutionary argument as more than a plausible mechanism. His bandit was an easy one, with the five options 0.15 apart in quality and the noise the same size, and he ended by naming the next experiment: shrink the gap, raise the noise, and see whether a swarm listening to only two neighbors "starts believing the first charismatic (tiny) dancer it meets."

This paper runs that experiment and several more. We asked four questions. Does the equivalence hold quantitatively, and where does it stop holding? What does the division by the average reward, the one new term in Maynard-Cross Learning, actually buy? How often does a finite swarm settle on the wrong option, and can it recover? And what happens when some of the bees do not dance honestly? Several of the answers were known from earlier work on robot swarms and voter models, and Section 4 says which. Section 2 sets out the claim, Section 3 says what "a single reinforcement learning agent" does and does not mean here, Section 4 covers earlier work, Section 5 gives the methods, Section 6 the results, and Section 7 what they imply for swarms of machines and for how large language models are trained.

2 The claim and the math behind it

The setting is a multi-armed bandit: a fixed set of options, each paying a noisy reward drawn from a distribution the learner cannot see. A learner's policy is a probability for each option, and it improves by trying options and adjusting toward the ones that paid. Cross Learning (Cross, 1973) is one of the oldest such rules. After trying option k and receiving a reward r_k between 0 and 1, it moves probability toward k in proportion to that reward:

CROSS LEARNING, ONE SAMPLE

$$\pi_a \leftarrow \begin{cases} \pi_a + \alpha r_k (1 - \pi_a) & \text{if } a = k \\ \pi_a - \alpha r_k \pi_a & \text{otherwise} \end{cases}$$

k is the option tried, $r_k \in [0, 1]$ its reward and α the learning rate. Averaged over which option is tried and what it pays, each probability moves by $E[\Delta\pi_a] = \alpha\pi_a(q_a - v)$, where q_a is option a 's expected reward and $v = \sum_b \pi_b q_b$ the policy's average reward. With a small learning rate, that is the Taylor replicator dynamic of evolutionary game theory (Taylor & Jonker, 1978; Börgers & Sarin, 1997).

The weighted voter model (Valentini, Hamann & Dorigo, 2014) is a simplified account of how honey bees choose a nest. Each scout holds one site. It estimates that site's quality, dances for it at a rate proportional to the estimate, and switches to the first dance it perceives from the bees around it. No bee ever sees another bee's estimate, only its dance. Soma et al. show that an infinite, well-mixed population following this rule changes its shares by the Maynard-Smith replicator dynamic, which is the Taylor form divided by the average:

TWO REPLICATOR DYNAMICS

Taylor: imitation of success, and Cross Learning in expectation

$$\frac{d\pi_a}{dt} = \pi_a(q_a - v)$$

Maynard-Smith: the weighted voter, and Maynard-Cross Learning in expectation

$$\frac{d\pi_a}{dt} = \frac{\pi_a}{v}(q_a - v)$$

Here π_a is the share of the population on option a . The only difference between the two is the division by v , the population's average quality (Soma et al., 2024, v1 eqs. 3 and 4).

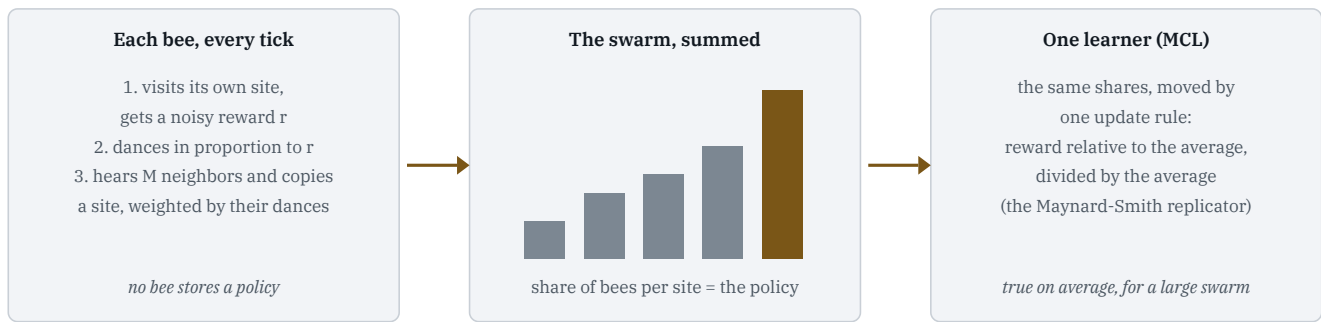
and that a single learner applying the matching per-sample update, which they name Maynard-Cross Learning, follows the same dynamic in expectation:

MAYNARD-CROSS LEARNING, ONE SAMPLE

$$\pi_a \leftarrow \begin{cases} \pi_a + \alpha \frac{r_k}{v}(1 - \pi_a) & \text{if } a = k \\ \pi_a - \alpha \frac{r_k}{v}\pi_a & \text{otherwise} \end{cases}$$

$\alpha = 1/N$: the learning rate is one over the number of bees, and batching the update over many samples corresponds to a larger swarm (Soma et al., 2024, v1 §3.2). The learner cannot see v , so it estimates it: with a moving average of recent rewards when it updates one sample at a time, and with the batch's mean reward when it updates in batches (v1 §4.2).

The practical reading, which Tanner draws out, is that each bee is a sampler running in its own copy of the environment and the swarm is the learner. The paper's later version adds simulations showing that about 500 scouts are enough to follow the replicator dynamic closely on its test problem, that a neighborhood of five or more bees is enough to approximate the well-mixed assumption, and that with a neighborhood of one "each scout copies an arbitrary neighbor, which yields no macro-dynamic" (Soma et al., 2024, v4 §§4.2-4.3; the authors add that their swarm-size figure "should be interpreted qualitatively rather than quantitatively"). Neither version models dishonest bees or deliberate exploration, and options whose quality changes over time appear only as other researchers' related work (v4, appendix B).



The claim under test: the middle panel moves exactly as the right panel says it should.

Figure 1. The two readings of one swarm. Each bee follows a local copying rule and stores nothing; the share of bees on each site is the policy; and the equivalence says those shares move, on average, the way a single Maynard-Cross learner's probabilities would. **Schematic.**

3 How a reinforcement learning agent differs from the hive

"The hive is a single reinforcement learning agent" is true in a narrow and exact sense: the swarm's shares move the way the simplest reinforcement learner's probabilities move, on average, for a large swarm, on a one-shot choice among a few options. Beyond that the resemblance ends, and the paper's own conclusion says so, contrasting reinforcement learning's "single-agent perspective, where information from consecutive action samples/batches is accumulated into one centralized agent's policy" with the swarm's "multi-agent perspective" and "emergent collective policy" (Soma et al., 2024, v1 §6). Table 1 sets the three side by side.

Table 1. The hive, the reinforcement learning agent it is equivalent to, and a full reinforcement learning agent of the kind that flies a drone or is used to train a language model (Sutton & Barto, 2018).

	The hive	The equivalent learner (a bandit)	A full RL agent
Where the policy lives	Nowhere. It is the head count of bees per site.	One probability per option, stored in one place	A table or a neural network, stored in one place
Who learns	No one. Each bee copies; the update exists only in the sum.	One learner applies one rule	One learner, usually by gradient on the network's weights
Samples per step	N at once, one per bee	One, or a batch; a batch is a bigger swarm	Whatever the designer chooses
How exact	On average, and only for a large swarm	Exactly what the code says	Exactly what the code says
Memory	What each bee prefers now; nothing else	A running estimate of the average reward	Values of states and actions, sometimes a model of the world

	The hive	The equivalent learner (a bandit)	A full RL agent
Time	One choice, no later consequences	Stateless: try, get paid, done	States and delayed rewards: which earlier move earned today's reward
Exploration	Only through bees still on other sites; an abandoned site is gone	The same, unless added	Built in on purpose
Where reward comes from	Each bee's own report, its dance, which nothing checks	The environment, trusted by construction	The environment, a learned judge, or a checker
Losing one part	Costs $1/N$; there is no center	A single point of failure	A single point of failure

The rows that matter for what follows are exactness, exploration and the reward channel. The equivalence is a statement about averages over an infinite population; a real swarm is finite, so it drifts. The swarm explores only through bees that have not yet switched, so once every bee has left a site, no bee will ever visit it again. And the swarm learns from its members' own reports of what they found, which a single learner never has to trust. Each of those is tested below.

4 What was already known

Several of our results have close precedents, and what follows should be read against them. The weighted voter model comes from Valentini, Hamann and Dorigo (2014), who studied it with a mean-field model, a master equation and agent-based simulations of a choice between two sites. They found that its accuracy "depends positively on the size of the swarm: Bigger swarms are more accurate," that accuracy did not depend on how far agents could communicate while the time to decide did, and that the model was robust to noisy estimates of quality, and they gave a minimum swarm size for a required accuracy. Our Sections 6.2, 6.5 and 6.9 restate those findings for five options with uniform reward noise; they are not new phenomena. The next year the same authors compared the weighted voter with a majority rule and concluded that the majority rule decides much faster at some cost in accuracy, while the weighted voter "takes much longer to establish a decision but guarantees the optimal solution" in the mean-field limit (Valentini, Hamann & Dorigo, 2015). The source paper's later version adds that a swarm with too few scouts "often yields convergence to a sub-optimal nest site option" (Soma et al., 2024, v4 §4.2), a figure we reproduce in Section 6.1.

Why a small swarm settles on a worse site is the population-genetics result on fixation: in a small population, chance can carry a slightly worse type to fixation, and what decides it is the population size times the relative advantage (Kimura, 1962). For this model the relative advantage per tick is the gap between two sites divided by the average reward, so the natural rule for swarm size is $N \times \text{gap} / v$, not $N \times \text{gap}$. Section 6.5 tests that form directly.

Dishonest members of a swarm are well studied. In the voter model, a single "zealot" who favors one opinion can pull an infinite population to unanimity in one and two dimensions, though not in higher

ones (Mobilia, 2003). In robot swarms choosing the best of several options, Canciani, Talamali, Marshall and Reina (2019) added "sects" of zealots that "keep communicating a constant opinion for a (possibly) inferior option" and, like all their attackers, do not modulate how often they communicate by quality but broadcast "every timestep." Five coordinated zealots in a swarm of 100 were enough to make most of the strategies they tested choose a much worse option; strategies based on a majority rule or on cross-inhibition held up best. Strobel, Castelló Ferrer and Dorigo (2018) modeled a Byzantine robot that "always votes for the minority color" and "keeps a quality estimate of" 1.0, found that the classical strategies, run to the end, converged to the wrong color, and defended with a blockchain contract that identifies and excludes such robots. The liars in Section 6.7 are the same adversary as in these two studies: they never switch, and they always advertise at full strength.

Finally, the two replicator dynamics in Section 2 differ at every instant by a positive factor, $1/v$. Multiplying a flow by a positive function changes how fast it moves along its paths but not the paths, so the Taylor and Maynard-Smith forms pass through the same sequence of policies and end in the same place; only their clocks differ. The comparison of rescaled and shifted rewards in Section 6.4 follows from that, and we present it as a consequence rather than a discovery.

Against this background, what is new here, as far as our search found, is: the size of the finite-swarm effects for the Maynard-Cross equivalence in particular, with the swarm-size rule stated in its dimensionless form and tested by shifting rewards; a test that separates the two things the standard adversary does, refusing to switch and exaggerating; a defense in which a bee checks the site itself rather than weighing a report, set against the source paper's question of why bees do not compare; and the link to how language models are trained with reinforcement learning. The search behind this section was not systematic, and we may have missed relevant work.

5 Methods

We simulated the weighted voter model directly. Each tick, every bee draws a reward for its own site, uniform in $[q - \delta, q + \delta]$ and clipped to $[0, 1]$; hears M bees drawn at random from the swarm (or, where M is "all," the whole swarm); and adopts a site with probability proportional to the rewards those bees drew for their sites. The swarm's share on each site is the policy. The paper's own simulations implement the choice centrally (v1, footnote 12); ours give every bee its own random neighborhood. Five bandits were used (Table 2). Each experiment ran a fixed range of random seeds starting at 0, and every configuration wrote one record, with its parameters, seeds, metrics and the hash of the script, to a log from which every number in this paper is read.

Table 2. The bandits. Five options each; δ is the half-width of the uniform reward noise.

Name	Option qualities	δ	Gap
Easy (Tanner's)	0.25, 0.40, 0.55, 0.70, 0.85	0.15	0.15
Medium	0.50, 0.55, 0.60, 0.65, 0.70	0.30	0.05

Name	Option qualities	δ	Gap
Hard	0.60, 0.62, 0.64, 0.66, 0.68	0.30	0.02
Very hard	0.60, 0.61, 0.62, 0.63, 0.64	0.30	0.01
Low (for the scaling test)	0.10, 0.15, 0.20, 0.25, 0.30	0.10	0.05

Three extensions were added for specific tests. *Scouting*: each tick, each bee re-picks a site uniformly at random with probability ϵ . *Liars*: a fixed set of bees start on one site, never switch, and always dance as if its reward were 1.0; to separate those two behaviors, Section 6.7 also runs *stubborn* bees, which never switch but dance their true reward, and *exaggerators*, which dance 1.0 while on the target site and their true reward elsewhere but copy other bees like anyone else. *Verify before switching*: a bee about to switch first visits the candidate site k times and its own site k times, and switches only if the candidate did at least as well. The metrics are the share of the swarm on each site, the tick at which the best site first holds 95% of the swarm, whether and where the whole swarm settled on one site ("fixation"), and, for the experiments with liars, the share of honest bees on each site averaged over the last 100 of 600 ticks.

To check the simulator against the source paper, we reimplemented its swarm-size experiment (v4 Fig. 2b) from the authors' public code rather than from our own: ten sites with qualities evenly spaced from 0.4 to 0.7, rewards uniform within ± 0.1 , the swarm starting split evenly across the sites, each bee weighing the dances of every other bee, 250 steps and 1,000 runs per swarm size. Where a result is a count of runs, we give a 95% Wilson interval; where it is a mean over runs, a 95% percentile bootstrap interval over the runs.

6 Results

6.1 Two baselines reproduce

Running Tanner's notebook unchanged gave every number the post reports (Table 3). We then ran the source paper's own swarm-size experiment as its public code sets it up. The authors seeded each run from the clock, so an exact match is not possible, but agreement within sampling error is: read from the vector data of the published figure, its curves end at 33.6%, 48.6%, 77.9%, 96.3% and 100% of the swarm on the best site for 10, 20, 50, 100 and 500 bees, and each lies inside the 95% interval of our 1,000 runs (Table 3). The experiments below start from two baselines anyone can check.

Table 3. Two baselines. Top: the post's reported numbers and ours, from the post's own code. Bottom: the source paper's Figure 2b (v4), the average share of the swarm on the best of ten sites after 250 steps, read from the published figure's vector data, against ours from 1,000 runs with a 95% bootstrap interval.

Quantity	Published	Measured
Single learner, share on the best arm after 2,500 steps, seed 42	0.952	0.952
Same, four seeds: range and mean	91.5% to 97.9%, 95.2%	91.5% to 97.9%, 95.2%

Quantity	Published	Measured
Hive, 8 neighbors: 95%, 99%, fixation	about 22, 24, 33	22, 24, 33
Hive, 2 neighbors: 95%, fixation	about 37, 58	37, 58
Source paper, 10 bees	33.6%	35.1% (32.3 to 37.9)
Source paper, 20 bees	48.6%	49.1% (46.1 to 52.3)
Source paper, 50 bees	77.9%	77.3% (74.7 to 79.9)
Source paper, 100 bees	96.3%	95.4% (94.0 to 96.6)
Source paper, 500 bees	100%	100%

6.2 The equivalence holds for a large swarm, and bends below it

On the medium bandit, the replicator dynamic reaches 95% on the best site at tick 41. A 1,000-bee swarm in which every bee hears the whole swarm reaches it at tick 41 too, and its mean over 100 runs never strays more than 0.015 from the replicator curve (Figure 2). Below that the fit loosens in two ways. Small neighborhoods are slower: with 1,000 bees each hearing two others, the swarm takes 70 ticks and strays by up to 0.21. Small swarms go the other way. Fifty bees reach 95% in 26 to 32 ticks, depending on the neighborhood, faster than the math, and in 12% to 25% of runs the whole swarm settles on a worse site. Those times are medians over the runs that reached the best site at all, so they flatter the small swarm. The speed is chance doing the converging, not selection. The paper's later version reports the same slowing for small neighborhoods (v4, appendix D.9), and Valentini et al. (2014) found that bigger swarms of this model are more accurate.

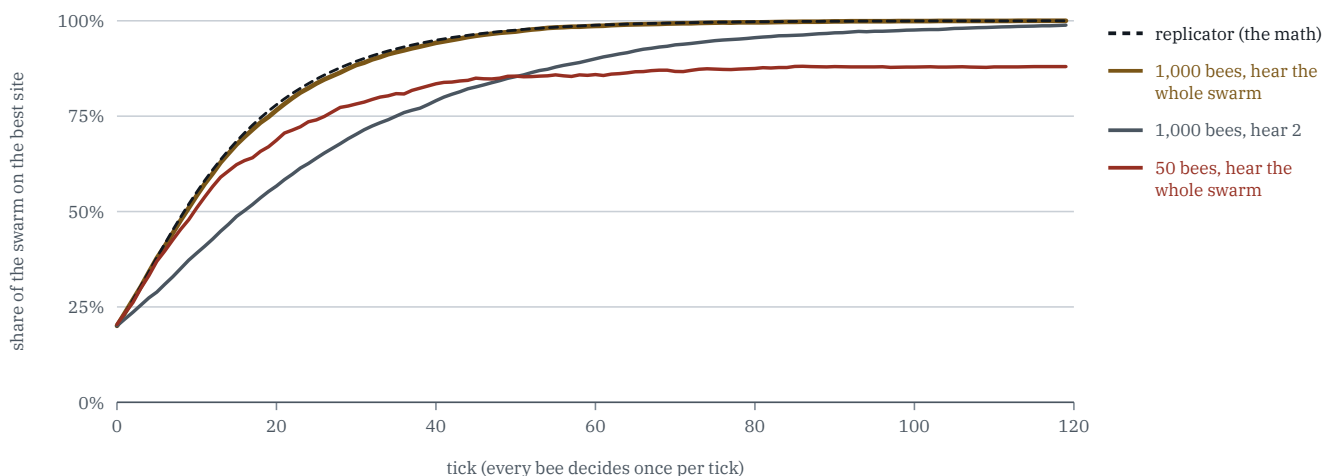


Figure 2. Mean share of the swarm on the best site, medium bandit, 100 runs per line. A large swarm hearing everyone sits on the replicator curve; hearing two neighbors is slower; a 50-bee swarm rises faster and levels off lower because some runs settle on the wrong site. **Simulated; seeds 0 to 99.**

6.3 Per reward sampled, the swarm is not smarter

Counting rewards drawn rather than ticks, the swarm needed 3,300 rewards to reach 95% on the easy bandit and 8,100 on the medium one (200 bees, 8 neighbors, medians of 30 runs, without intervals). The serial Maynard-Cross learner from the post needed 2,849 and 7,157, and a serial Cross learner 3,619 and 11,486. The swarm got there in 16.5 and 40.5 ticks because it draws 200 rewards per tick. Its advantage is entirely that it has many bodies working at once, which is Tanner's point about clocks, now with numbers attached. Part of Maynard-Cross Learning's lead over Cross Learning here is simply step size: with rewards averaging about 0.6, dividing by the average makes every step about 1.7 times larger.

6.4 What dividing by the average buys, and what it costs

Tanner suggests deleting the division to see what it is for. Section 4 gives the short answer: dividing by v changes the clock, not the path. We ran the weighted voter (Maynard-Cross) and imitation of success (Cross Learning) on the same low-reward bandit with every reward rescaled or shifted (Figure 3; 500 bees, 100 runs each). Rescaling every reward leaves the weighted voter's vote probabilities exactly as they were, so the same seeds gave the same runs, 18 ticks to 95% at every scale; that is a check of the code against the algebra, not a measurement. Adding 0.3 or 0.6 to every reward slowed it to 37 and 55 ticks, because a shift shrinks every option's advantage relative to the average. Imitation of success is the mirror image: 56.5, 114 and 277.5 ticks under the rescaling, and 56.5 and 52.5 under the shifts. Dividing by v buys immunity to the scale of rewards and gives up immunity to their offset. The paper's own analysis anticipates the first half: it shows the Maynard-Smith dynamic's speed is at least the Taylor dynamic's, with the difference largest when rewards are near zero (v_1 §5). For an infinite swarm this is only a matter of speed. For a finite one it is not, because selection slows under a shift and drift does not; Section 6.5 measures what a shift does to accuracy.

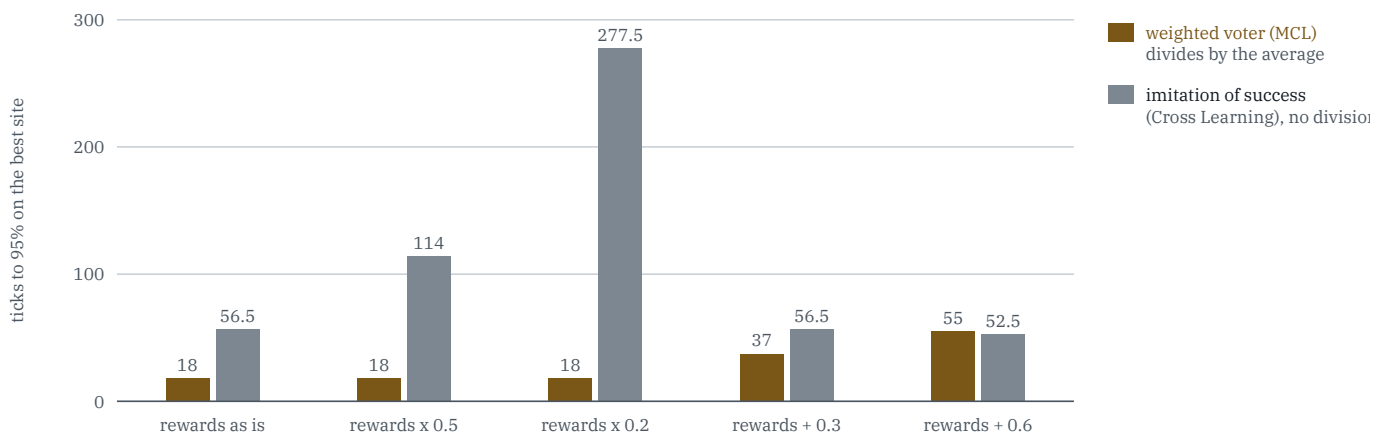


Figure 3. Median ticks to 95% on the best site under rescaled and shifted rewards, low bandit, 500 bees, 100 runs per bar. **Simulated; seeds 0 to 99.**

6.5 How often a swarm settles on the wrong answer

Across three gaps, five swarm sizes and three neighborhood sizes, 200 runs each, every run fixated, and the share that fixated on a worse site fell with swarm size (Table 4), as Valentini et al. (2014) found for two options. Swarm size matters more than neighborhood size. In the eight cells where both rates were above 1%, hearing two neighbors instead of eight raised the error rate in seven, by 1.3 to 4.3 times, and tied in the eighth (52% against 51%); hearing the whole swarm instead of eight helped less than growing the swarm did. Real swarms field 200 to 500 scouts out of roughly 10,000 bees (Soma et al., 2024, v4 §4.2). In our noise model, 200 bees separated options 0.05 apart in all 600 runs across the three neighborhoods, and options 0.02 apart in 89.5% to 97% of runs; 500 bees separated options 0.02 apart in all but one of 600 runs, and options 0.01 apart in 94% to 99.5%.

The first way to state the threshold is the number of bees times the quality gap, which reaches about 10 at 200 bees for a gap of 0.05, 500 for 0.02 and 1,000 for 0.01. That form has units: it moves with the scale and offset of the rewards. Section 4 gives the dimensionless form, $N \times \text{gap} / v$. We tested it with a prediction written into the code before the run: on the hard bandit, where the average reward starts at 0.64, adding 0.3 to every reward should raise the swarm needed by $(0.64 + 0.3) / 0.64 = 1.47$ times, and adding 0.6 should raise it by 1.94 times. The runs agree (Figure 4; 400 runs per point). At 200 bees, shifting every reward raised wrong settlements from 4.3% to 10% and to 15.8%. Plotted against $N \times \text{gap} / v$ instead of N , the three curves fall on one another: 16.8%, 14.8% and 15.8% wrong at values between 3.1 and 3.3, and 1.3%, 1.0% and 0.8% between 8.5 and 9.7. The unshifted medium and very hard bandits, with gaps of 0.05 and 0.01, fall close to the same curve. With eight neighbors, no run locked onto a worse site once $N \times \text{gap} / v$ reached 12.5: none in 2,400 runs across the three shifts, a 95% interval of 0 to 0.16%. With two neighbors the bar is higher: in Table 4, two cells near 16 each still had one wrong run in 200. The constant depends on the noise and the number of options, but the form does not depend on how rewards are scaled or shifted. Halving the gap still doubles the swarm you need.

Table 4. Share of 200 runs in which the whole swarm settled on a worse site. M is the number of neighbors each bee hears; "all" is the whole swarm. With 200 runs a cell, a 95% Wilson interval for 0 is 0 to 1.9%; for 0.5%, 0.1 to 2.8%; for 3%, 1.4 to 6.4%; for 10.5%, 7.0 to 15.5%.

Gap	M	50 bees	100	200	500	1,000
0.05	2	25.5%	6.5%	0	0	0
	8	12.5%	1.5%	0	0	0
	all	12.5%	0.5%	0	0	0
0.02	2	46%	31%	10.5%	0.5%	0
	8	35.5%	19%	4.5%	0	0
	all	34.5%	15.5%	3%	0	0
0.01	2	52%	51%	31%	6%	0.5%

Gap	M	50 bees	100	200	500	1,000
	8	51%	38.5%	21%	0.5%	0
	all	42.5%	34.5%	12%	0.5%	0

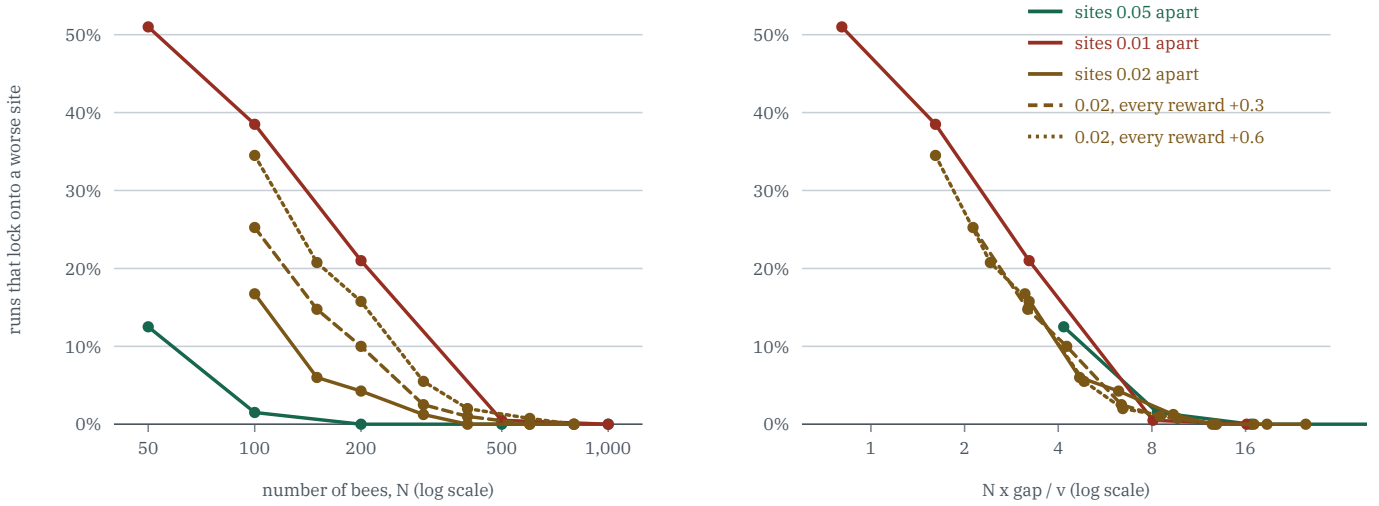


Figure 4. Runs that settle on a worse site, 8 neighbors. Left, by swarm size: closer sites, and a shift added to every reward, both raise the swarm needed. Right, the same runs by $N \times \text{gap} / v$: the curves collapse onto one. **Simulated; 200 runs per point (medium, very hard), 400 (hard, shifted).**

6.6 Once wrong, wrong for good, unless some bees scout

Pure copying has no way back to a site that no bee holds, so a wrong fixation is permanent. We added a small scouting rate ϵ to the hard bandit with 200 bees. With none, 4% of runs sat on the second-best site for all 1,500 ticks. At $\epsilon = 0.002$, meaning two bees in a thousand picking a site at random each tick, the best site led in every run and held 96.6% of the swarm; the scouting itself costs the rest. At 0.01 it held 84%. At 0.05, too much scouting swamped the selection: the best site held under half the swarm in 85% of runs. Starting every bee on the second-best site, no run recovered without scouting, and every run recovered with it, in a median of 260.5 ticks at 0.002 and 107.5 at 0.01.

The harsher test is a world that changes. On the medium bandit, we reversed the qualities at tick 200, so the best site became the worst and the worst the best (Figure 5). Without scouting, the new best site had no bees and never got any. With 1% scouting, the swarm moved to it in a median of 22.5 ticks and held about 93% there by tick 800; with 5% it moved as fast and held about 70%.

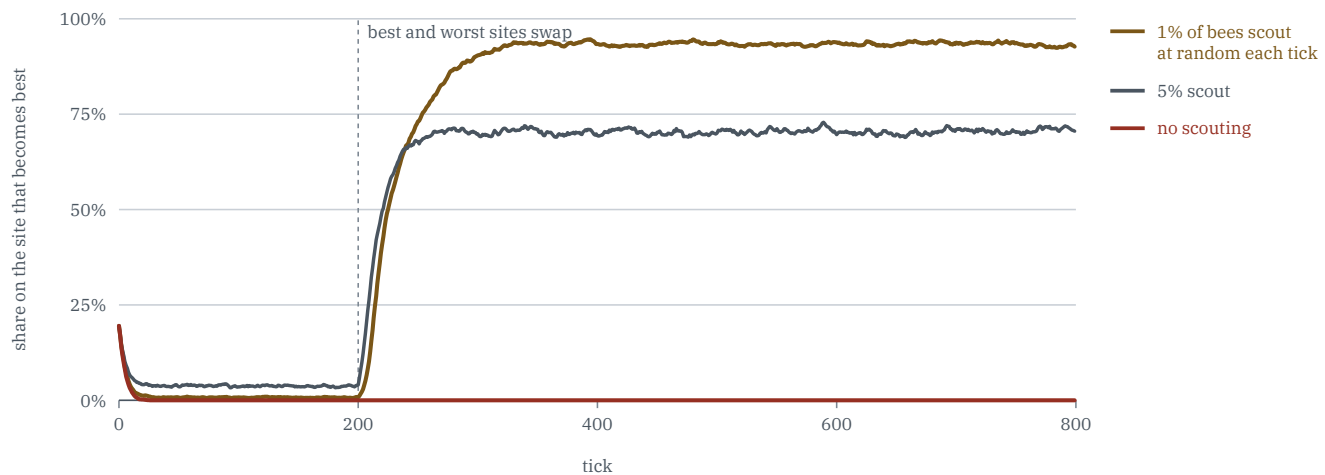


Figure 5. Share of a 200-bee swarm on the site that becomes the best when the qualities reverse at tick 200, medium bandit, 8 neighbors, 100 runs per line. **Simulated; seeds 0 to 99.**

6.7 A few stubborn bees decide a close contest

The weighted voter rule is simple because it asks nothing of a bee except to dance and to copy, and that same simplicity means it trusts every dance. We added liars of the kind studied by Canciani et al. (2019) and Strobel et al. (2018), bees that hold one site forever and always dance at full strength, to a 200-bee swarm hearing eight neighbors, and measured where the honest bees ended up (Table 5, Figure 6). On the easy bandit the damage is proportional: in a first pass (60 runs), 5% liars for the worst site held 9% of the honest swarm there. On the hard bandit it is not. Five percent liars for the worst site pulled 95% of the honest swarm onto it (a 95% interval of 92% to 97%), and 2% liars for the second-best site pulled 99%. The most effective lie is for an option almost as good as the best, because honest bees that follow it find nothing wrong enough to leave.

These liars do two things at once: they never switch, and they exaggerate. We separated the two (Table 6; 100 runs per cell), and stubbornness does most of the damage. Bees that never switch but dance their true reward, 2% of the swarm on the hard bandit's second-best site, still pulled 98% of the honest swarm there; for the worst site, 5% of them pulled 56%, against 95% when they also exaggerated. Exaggeration without stubbornness did little. Bees that dance at full strength on the target site but can be recruited away drew no honest bees to the worst site at all, because they were recruited away first. For the second-best site they mattered only at 5% and 10% of the swarm, where some runs went their way and some did not (44% and 99% of honest bees on average). Close to a tie, the weighted voter behaves like the plain voter model, in which a single zealot can carry a population (Mobilia, 2003). Tanner asked whether a swarm would believe the first charismatic dancer it met. In this model, a dancer who exaggerates but can be persuaded is mostly harmless; one who cannot be persuaded is the danger, even when it reports honestly.

Checking before copying defends against the liars in Table 5, and how well depends on how different the options really are. When a bee about to switch first visited the candidate site and its own site once each, the worst-site lie at 5% fell from 95% to about 10%; with five visits each, to about 3%. Against liars

for the second-best site, even five visits left 21% of the honest swarm on it at 5% liars and 44% at 10%, because a few noisy visits cannot reliably tell 0.66 from 0.68. The cost is in visits: the defended swarm drew up to 1,063 rewards per tick against the undefended swarm's 180 to 200. At the same number of visits against the same liars, the hard bandit cost more than the medium one in 11 of 12 settings, because more bees were still switching; the exception is a swarm the liars had already captured, where almost no one switches. Checking also fixed the honest failure from Section 6.5: with no liars at all, one visit raised the best site's share on the hard bandit from 95% to 100%.

Table 5. Share of honest bees on the liars' site, averaged over the last 100 of 600 ticks, 60 runs per cell. Liars never switch and always dance at full strength. "Verify k" means a bee visits the candidate and its own site k times each before switching. Brackets give a 95% bootstrap interval where it is wider than 1.5 points either side; elsewhere it is narrower.

Bandit	Liars' site	Liars	No defense	Verify 1	Verify 5
Medium	Worst	2%	11%	1%	0.1%
		5%	29%	3%	0.3%
		10%	61% [59, 64]	6%	0.5%
Medium	Second best	2%	57% [52, 62]	6%	2.5%
		5%	100%	17%	6.5%
		10%	100%	35%	14%
Hard	Worst	2%	31% [29, 34]	4%	1%
		5%	95% [92, 97]	10%	3%
		10%	100%	21%	7%
Hard	Second best	2%	99%	18%	8%
		5%	100%	50% [47, 54]	21%
		10%	100%	100%	44%

Table 6. The liar split into its two behaviors, no defense, 100 runs per cell. "Both" repeats Table 5's liars with 100 runs instead of 60. "Stubborn only" bees never switch but dance their true reward. "Exaggerating only" bees dance at full strength while on the target site and their true reward elsewhere, and copy other bees like anyone else. Brackets as in Table 5.

Bandit	Liars' site	Liars	Both	Stubborn only	Exaggerating only
Medium	Worst	2%	11%	6%	0
		5%	28%	15%	0
		10%	61% [60, 63]	33%	0

Bandit	Liars' site	Liars	Both	Stubborn only	Exaggerating only
Medium	Second best	2%	55% [52, 59]	34% [32, 36]	0
		5%	100%	98% [96, 99]	1% [0, 3]
		10%	100%	100%	10% [5, 16]
Hard	Worst	2%	30% [29, 32]	19%	0
		5%	95% [93, 97]	56% [53, 60]	0
		10%	100%	100%	0
Hard	Second best	2%	99.6%	98% [96, 99]	9% [4, 15]
		5%	100%	100%	44% [34, 54]
		10%	100%	100%	99% [97, 100]

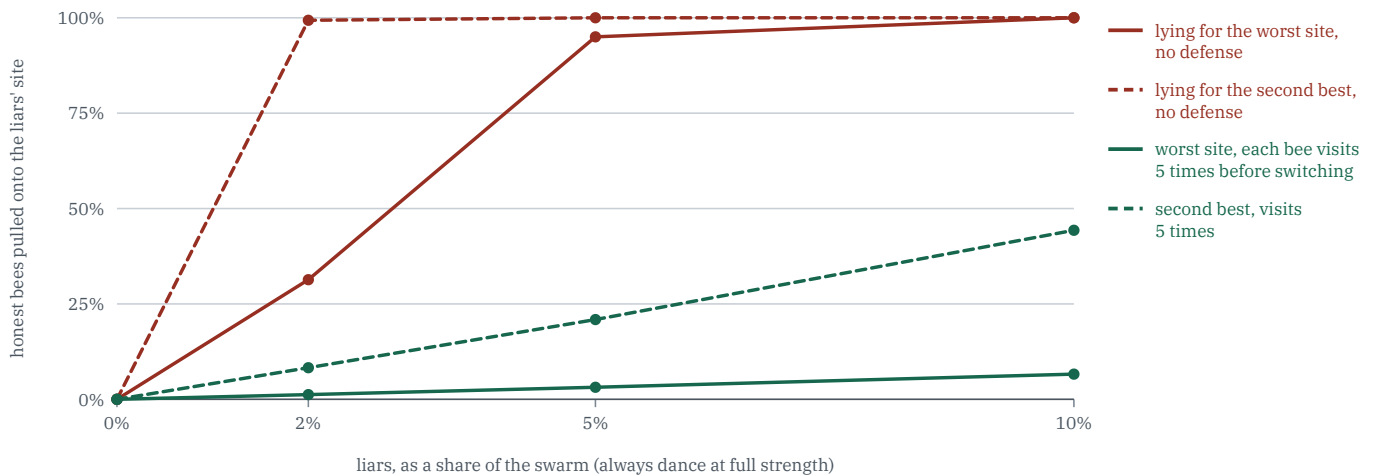


Figure 6. Honest bees pulled onto the liars' site, hard bandit, 200 bees, 8 neighbors. Solid lines: liars for the worst site; dashed: liars for the second best. **Simulated; 60 runs per point.**

6.8 Checking the site, not the report

The paper's later version asks why bees evolved blind copying when variants that compare quality estimates converge faster, and suggests the answer is that comparing requires a calibrated, shared scale for quality, which the weighted voter avoids (v4 §4.5). Our results add a second consideration. Comparing a peer's reported number would not have helped against the liars in Table 5, since the report is the lie. What helped was a bee checking the site itself. A deployed swarm that needs to resist bad actors needs that form of comparison, whatever it costs in speed. The robot-swarm defenses in Section 4 work differently: a majority rule or cross-inhibition limits how much any one message counts (Canciani et al., 2019), and a blockchain contract identifies and excludes the robots whose votes break

its rules (Strobel et al., 2018). We did not compare those defenses with checking the site, and we did not test checking against the stubborn but honest bees of Table 6, whose reports are true. Both are the natural next experiments.

6.9 Physical neighbors cost time, and no accuracy we could detect

Bees and robots hear the individuals near them, not a random sample of the swarm. We placed bees on a ring and let each hear its eight nearest neighbors. On the medium bandit, every run still reached the best site, but consensus took about twice as long (95% at a median of 76 ticks against 43.5 for a well-mixed swarm of 200; 89 against 45 for 1,000). On the hard bandit, 100 runs each, the ring was 2.3 to 3.4 times slower (203.5 against 88.5 ticks at 200 bees; 361.5 against 107.5 at 1,000). Its accuracy showed no difference: 4 wrong fixations in 100 runs at 200 bees either way, each with a 95% interval of 1.6% to 9.8%, and none at 1,000. That test could not have detected a difference of a few points. Valentini et al. (2014) likewise found that the reach of communication changed the time to decide but not the accuracy.

7 What it means

7.1 The equivalence, read carefully

The result stands, and it is useful. A large swarm of copiers does move the way a single Maynard-Cross learner does, closely enough that its trajectory is indistinguishable from the replicator curve at 1,000 bees. What the equivalence does not carry over from the single learner is the single learner's guarantees. A finite swarm drifts, and drift can finish the job before selection does. A swarm with no scouting cannot explore, so it cannot recover from a wrong settlement or follow a change. And a swarm learns from its members' reports, so its accuracy is only as good as their honesty. These are not objections to the paper, which makes no claims about any of them, and the first and third were known from earlier work on this model and on robot swarms (Section 4); they are the terms on which the equivalence can be used.

7.2 The same design questions in how large language models are trained

Cross Learning does not appear by name in the published recipes for training large language models that we know of, and the reason is size. It assumes a handful of options and a table of probabilities; a language model's "option" is an entire response, and its policy is a neural network. The update that scales is its parametric relative, the policy gradient (Williams, 1992), in forms such as proximal policy optimization (Schulman et al., 2017). But the swarm's structure reappears. Group Relative Policy Optimization samples a group of responses to the same prompt, a small swarm, and scores each one against the group, subtracting the group's mean reward and dividing by its standard deviation (Shao et al., 2024). That is the question Section 6.4 tests: what to divide by. The group-relative form is immune to both the scale and the offset of rewards, and it is contested. Liu et al. (2025) show that dividing by the standard deviation gives extra weight to questions that are too easy or too hard, a "question-level difficulty bias," and remove it.

The swarm's three failures have counterparts in this training, by analogy rather than by any shared mechanism we tested. A policy that stops producing some type of answer can no longer learn that it was good, which is one reason training keeps the model near where it started, for example with "a per-token KL penalty from the SFT model at each token to mitigate over-optimization of the reward model" (Ouyang et al., 2022). A learned judge can be gamed the way the liars game the dance: optimizing a proxy reward model too hard makes the true objective worse (Gao, Schulman & Hilton, 2022). And one defense has the same shape as verifying before switching. DeepSeek-R1-Zero was trained with rule-based rewards that check answers, not with a neural reward model, because such a model "may suffer from reward hacking in the large-scale reinforcement learning process"; the released DeepSeek-R1 kept rule-based rewards for its reasoning data and added reward models for general data (DeepSeek-AI, 2025).

7.3 For anyone fielding a swarm

Tanner's case for swarms in defense and physical AI is that a flock of cheap, simple agents can search and commit with no central server to depend on or lose. Our results support that, and they point to four design rules, most of which the robot-swarm work in Section 4 would also suggest. Size the swarm to how close the options are relative to the average payoff: in our model, with each agent hearing eight others, the number of agents times the quality gap, divided by the average reward, needs to reach about 12.5, and the constant will differ with the noise, the number of options and the size of the neighborhood. Keep a small scouting rate, a few agents in a thousand, so that a wrong settlement can be undone and a change in the world can be followed. Make agents check a claim themselves before acting on it, because a swarm with no center also has no one checking the reports, and the cheapest attack is a small lie for a plausible option. And expect physical locality to slow consensus rather than corrupt it. None of these requires a central learner; each costs something the pure weighted voter rule does not pay.

7.4 A hypothesis for teams of AI agents

We did not test anything about AI agents, so this section is a hypothesis, not a finding. A team of AI agents that learns from its own work, reusing each other's code, notes and lessons, is also a population of copiers, and the same three failures could apply. A small team is closer to 50 bees than to 1,000, so which practices spread might depend more on chance than on merit when the practices differ only a little. A practice that everyone copies, with no one trying the alternative, could outlive the conditions that made it right. And a confident report may be copied whether or not anyone checked it. If so, the fixes would be the ones above: size the check to how close the calls are, keep some deliberate trial of alternatives, and have the copier check a claim against its source rather than trusting how strongly it was stated. Testing this would take a measured team, not a simulated swarm.

8 Limits

These are simulations of a simplified model, not of bees or of any fielded system. The bandits have five options with uniform noise, and the constant in Section 6.5's rule is specific to them; it would move with the noise and the number of options, although the tests there suggest the form $N \times \text{gap} / v$ would not. We did not compare our finite-swarm results with the master-equation model of Valentini et al. (2014), which would predict them for two options. Neighbors were drawn uniformly from the whole swarm except in Section 6.9, where the ring is only one form of locality. Qualities were fixed except in the reversal test. Liars always reported the maximum and never adapted; a smarter adversary would do at least as much damage. Verification was modeled as unbiased extra visits, which assumes visiting a site is possible and honest. The paper under test was read in full in its first version (v1, 23 October 2024, then titled "Bridging Swarm Intelligence and Reinforcement Learning"); its current version (v4, 6 October 2025) was read for the passages cited and searched for any treatment of dishonest individuals, exploration or changing environments. We used the authors' public code only to set up the reproduction in Section 6.1; every other experiment runs on our own simulator, and because our code is not public, no one else has reproduced our results. Section 6.3's comparison rests on medians of 30 runs without intervals. None of this work has been peer reviewed.

Disclosure. The experiments were designed under the author's direction and written and run by an AI agent (Claude, Anthropic, in Claude Code), which also assisted with drafting; the author is responsible for the content. Every number in the paper was read from the run log, and the figures were drawn from it by a script that records each plotted series. Soma et al. (v1) and Tanner's post were read in full; Soma et al. (v4) was read for the passages cited and searched in full text, and its public code was read to set up the reproduction in Section 6.1. Canciani et al. (2019) was read in full; Valentini et al. (2014, 2015) and Strobel et al. (2018) were read for the passages cited; Mobilia (2003) was read in abstract only. Each arXiv reference was resolved at arXiv on 30 September 2026 with its title and authors matched, and every passage quoted from Ouyang et al. (2022), Liu et al. (2025), DeepSeek-AI (2025), Valentini et al. (2014, 2015), Canciani et al. (2019), Strobel et al. (2018) and Soma et al. (2024) was checked against its text. The older references, including Kimura (1962), are cited for standard results and were not re-read for this paper. **Competing interests.** The author owns Wolfberg LLC. The work was not funded. **Code and data.** The simulation code, the run log (one record per configuration, with parameters, seeds, metrics and the script's hash) and the figure data are available from the author on request.

References

- Börger, T., & Sarin, R. (1997). Learning through reinforcement and replicator dynamics. *Journal of Economic Theory*, 77(1), 1-14.
- Canciani, F., Talamali, M. S., Marshall, J. A. R., & Reina, A. (2019). *Keep calm and vote on: Swarm resiliency in collective decision making*. Extended abstract, ICRA 2019 Workshop on Resilient Robot Teams: Composing, Acting, and Learning. cl.cam.ac.uk

- Cross, J. G. (1973). A stochastic learning model of economic behavior. *The Quarterly Journal of Economics*, 87(2), 239-266.
- DeepSeek-AI. (2025). *DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning*. arXiv:2501.12948. arxiv.org/abs/2501.12948
- Gao, L., Schulman, J., & Hilton, J. (2022). *Scaling laws for reward model overoptimization*. arXiv:2210.10760. arxiv.org/abs/2210.10760
- Kimura, M. (1962). On the probability of fixation of mutant genes in a population. *Genetics*, 47(6), 713-719.
- Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W. S., & Lin, M. (2025). *Understanding R1-Zero-like training: A critical perspective*. arXiv:2503.20783. arxiv.org/abs/2503.20783
- Maynard Smith, J. (1982). *Evolution and the theory of games*. Cambridge University Press.
- Mobilia, M. (2003). Does a single zealot affect an infinite group of voters? *Physical Review Letters*, 91(2), 028701. arXiv:cond-mat/0304670. arxiv.org/abs/cond-mat/0304670
- Ouyang, L., Wu, J., Jiang, X., et al. (2022). *Training language models to follow instructions with human feedback*. arXiv:2203.02155. arxiv.org/abs/2203.02155
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal policy optimization algorithms*. arXiv:1707.06347. arxiv.org/abs/1707.06347
- Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., et al. (2024). *DeepSeekMath: Pushing the limits of mathematical reasoning in open language models*. arXiv:2402.03300. arxiv.org/abs/2402.03300
- Soma, K., Bouteiller, Y., Hamann, H., & Beltrame, G. (2024). *The hive mind is a single reinforcement learning agent*. arXiv:2410.17517 (v4, 6 October 2025; v1, 23 October 2024, titled *Bridging swarm intelligence and reinforcement learning*). arxiv.org/abs/2410.17517. Code: github.com/MISTLab/HiveMindRL (read at commit 43865ddcbf8c).
- Strobel, V., Castelló Ferrer, E., & Dorigo, M. (2018). Managing Byzantine robots via blockchain technology in a swarm robotics collective decision making scenario. In *Proceedings of AAMAS 2018* (pp. 541-549).
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press.
- Tanner, T. C., Jr. (2026, September 30). *SnakeByte[23]: Maynard-Cross Learning (MCL): Hive minds as a single RL agent*. tedtanner.org
- Taylor, P. D., & Jonker, L. B. (1978). Evolutionary stable strategies and game dynamics. *Mathematical Biosciences*, 40(1), 145-156.
- Valentini, G., Hamann, H., & Dorigo, M. (2014). Self-organized collective decision making: The weighted voter model. In *Proceedings of AAMAS 2014* (pp. 45-52).

- Valentini, G., Hamann, H., & Dorigo, M. (2015). Efficient decision-making in a self-organizing robot swarm: On the speed versus accuracy trade-off. In *Proceedings of AAMAS 2015* (pp. 1305-1314).
- Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8, 229-256.