Evolving Local Rules for Agent Societies: A Preliminary Study

AI
The Latest Models

Xisen Wang
University of Oxford
xisen.agi@gmail.com

Qi Zhang
University of Oxford
qi.zhang.agi@gmail.com

Abstract

How much collective intelligence can emerge from simple, decentralized rules without heavy, predefined workflows? We explore this question through a framework that couples Boids-style local coordination with explicit evaluation and selection in a survival-driven, tool-building ecology. Agents interact via three local rules—cohesion, separation, and alignment—and follow an observe–reflect–build loop to generate and refine tools within an ecosystem that includes automated tests, shared registries, and a Tool Complexity Index capturing code, interface, and compositional sophistication. Positioned as a preliminary study, this work treats evolution as a complementary lens: local rules catalyze collaboration and modularity, while selective feedback favors strategies that persist across generations. Across text analysis, data science, and simulation modeling tasks, evolutionary-Boids societies increase throughput and balance contributions among agents while maintaining reliability, though current prompting tends to suppress deep tool composition. The resulting systems produce more, smaller, and self-contained tools rather than extended pipelines, suggesting a breadth-first mode of exploration. Overall, the framework offers an early step toward understanding how simple, local interactions and evolutionary pressure together shape the emergence of organized, evolving agent ecosystems.

1 Introduction

How much collective intelligence can emerge from *simple, decentralized rules*—without heavy, predefined workflows? Multi-agent systems have revealed striking forms of coordination, communication, and division of labor, yet most contemporary setups still rely on task-specific scaffolds and rigid pipelines. What remains underexplored is whether minimal local interactions alone can catalyze collaboration and long-horizon adaptation in open-ended settings, and how such societies evolve, specialize, and govern themselves over time.

Foundational work on flocking showed that three local rules—separation, alignment, and cohesion—can yield sophisticated global structure without centralized control [1, 2, 3]. Similar principles appear across biological and engineered collectives [4, 5, 6, 7, 8]. In parallel, evolutionary systems demonstrated how variation and selection can drive continual change, from early artificial life platforms to diversity-seeking algorithms and autocurricula [9, 10, 11, 12, 13, 14**?**]. Despite this progress, gaps persist: swarm models typically lack long-term adaptation, evolutionary approaches can stagnate prematurely, and emergent coordination is often bounded by narrow, workflow-centric tasks [15, 16, 17, 18].

We argue that a unifying perspective is to decompose decentralized agent collaboration into *local rules* plus an *evolution algorithm*. Local rules shape how agents interact and adopt each other's artifacts; evolutionary evaluation selects which behaviors and artifacts persist. Tool building is a particularly revealing lens here: while tool *use* by LLMs and agents is well studied [19, 20, 21, 22, 23], the

*These authors contributed equally to this work.

collaborative creation and refinement of tools exposes modularity, composability, and ecosystem dynamics that are hard to observe in single-task workflows.

We introduce a preliminary framework that couples Boids-style local coordination with explicit evaluation and selection in a survival-driven, tool-building ecology. Agents follow an *observe–reflect–build* loop to create and refine tools; the ecosystem provides automated tests, shared registries, and a *Tool Complexity Index* that quantifies code, interface, and compositional sophistication. Boids-inspired rules encourage decentralized coordination (e.g., adoption without central planning and functional specialization), while evolutionary pressure serves as a complementary lens to study longitudinal adaptation, retention, and collapse. Our results show that Evolutionary-Boids has the potential to reliably increase throughput and balance contribution across agents.

The paper is organised as follows. We begin by presenting the baseline system, followed by the design of the agent society motivated by tool construction. Next, we introduce the computational framework based on Boids dynamics, and subsequently describe the evolutionary algorithm that governs system adaptation. Finally, a small-scale empirical study is conducted to validate the framework and demonstrate its potential.

2 Related Work

Local Interaction Rules, Coordination, and Emergent Intelligence. Classical results demonstrate that simple local interactions can produce coherent global structure without centralized control. Reynolds' *Boids* established that separation, alignment, and cohesion suffice for lifelike flocking [1], while statistical physics models proved long-range order and nonequilibrium phase transitions in self-propelled particles [2, 3]. Biology and crowd dynamics provide convergent evidence that decentralized feedbacks and attractive/repulsive "social forces" yield large-scale coordination [4, 5], and control-theoretic and swarm-engineering work formalizes distributed flocking with design and verification principles [6, 7]. Behavioral ecology further links local cues to collective decisions and leadership [8]. We adopt this micro-to-macro lens but recast alignment/cohesion/separation as *institutional primitives* subject to evolutionary pressure in survival-driven ecologies.

Evolutionary algorithms for open-ended adaptation. Digital evolution showed that variation and selection can sustain innovation and coevolution in silico [9, 10]. To mitigate deception and premature convergence, novelty search and quality–diversity (QD) maintain behaviorally diverse, high-performing repertoires [11, 12, 13], with repertoire-based control enabling rapid self-recovery in robotics [24]. Open-ended approaches co-evolve challenges and solutions via transfer across stepping stones (POET and variants) [14], while unsupervised environment design induces curricula that yield robust zero-shot transfer [25]. We adopt this diversity-first view but define fitness at the *societal* level: evolution acts jointly on agent policies and the institutional/tool layer, retaining strategies and rules that improve collective performance and stability.

Open sandbox simulations with a slice toward tool *creation*. Open multi-agent sandboxes probe social generalization and emergent dynamics at scale: self-play yields staged strategies and *emergent tool use* [15]; XLand trains generally capable agents across procedurally generated social tasks [17]; Melting Pot 2.0 targets novel-partner generalization in mixed-incentive settings [26]; Neural MMO 2.0 offers persistent many-agent worlds with multi-task evaluation [18]; and Overcooked-based setups benchmark zero-shot human–AI coordination and layout generalization [16, 27]. Complementary LLM-agent work studies how *tools* and *skills* are acquired and orchestrated: Toolformer learns API calling [19]; Voyager accumulates persistent embodied skill libraries [20]; multi-agent scaffolds (CAMEL, AutoGen) coordinate role-specialized LLMs [21, 22]; and "generative agents" simulate long-horizon social behavior [23]. Reviewer-authored systems extend this frontier—*Agent LUMOS* (modular training) [28], *OASIS* (scaling to one million agents) [29], *OWL* (hierarchical multi-agent workforce) [30], and schema-guided, culture-aware role-play [31]—while *CollabUIAgents* analyzes credit re-assignment for collaboration and generalization [32]. Our contribution is a minimalist, Boids-style *survival-driven sandbox* in which agents not only *use* tools but also *create* and *retain* tools and rules, with evolutionary selection determining which institutions persist or collapse.

3 Methodology

3.1 Baseline System: Self-Reflective Tool-Building Agent Society

Overview and agent loop. Our baseline establishes the minimal viable setting in which decentralized agents generate and share tools while collective structure emerges. Each agent follows a simple observe–reflect–build loop grounded in five conceptual components: an Agent Identity with a light specialization prior; a Shared Tool Registry that records community-visible artifacts; a Personal Tool Space for private development and testing; a Reflection History logging prompts, generated reflections, and bookkeeping metadata; and an Environment Manager abstracting resources and constraints. At each timestep, the agent inspects available tools and their test outcomes and proposes a new tool. Tools expose a standardized interface that enables composition—simple primitives combine into larger workflows—executed in a centralized context guarded by recursion-depth limits. Adoption signals are approximated by static scans of promoted tools for references. This compositional substrate encourages dependency chains across agents and provides the basic medium for emergent collaboration.

Assurance and specialization dynamics. Every tool proposal triggers smoke-test–oriented quality control comprising autogenerated test harnesses (candidate calls validating execution without exceptions), execution outcomes (pass/fail and error logs), visibility (propagating outcomes of promoted, passing tools to all agents; failing tools remain private), and persistence (structured logs for longitudinal study). These mechanisms steer the ecosystem toward reliability without claiming deep functional coverage. On top of this, we incorporate light biases that promote division of labor: Meta-Prompt Influence nudges agents toward broad domains without hard constraints; Usage-Based Reinforcement is realized by prioritizing more widely adopted tools when presenting exemplars back to agents; Alignment-conditioned guidance is emitted only when higher-producing, successful neighbors exist; no explicit failure-driven branch is enacted. Together, assurance and bias produce a feedback loop in which successful tools become more visible, unsuccessful ones are inspected and revised, and niches of specialization gradually crystallize.

Infrastructure, observables, and study design. All experiments are designed to run in isolated, reproducible executions that could emit structured logs of reflections, tool creations, and evaluations, together with quantitative traces in JSON for potential post-hoc analysis and rich console visualizations to monitor ecosystem dynamics. We propose a set of observables that could be used to summarize emergent behavior in future studies: Tool Creation Rate (new tools per agent per round), Composition Depth (average dependency-chain length), Specialization Index (diversity of tool types across agents), Collaboration Events (frequency with which tools build on others), Test Success Rate (ecosystem reliability), and adoption trends estimated from static dependency scans. This proposed instrumentation outlines how experimental control and comparability could be achieved across conditions, establishing a quantitative foundation on which communication protocols and evolutionary pressures may later be layered to assess their impact on coordination, specialization, and long-horizon performance.

3.2 Computational Framework for Boids-Inspired Cognitive Coordination

Our framework adapts the classical boids model from spatial coordination to the cognitive domain of multi-agent tool creation. The core of an agent's decision-making process is governed by three rules—separation, alignment, and cohesion—which we formulate mathematically to guide behavior based on local information within the agent's neighborhood. In the present implementation, these rules act as *prompt-level conditioning signals*, and the orchestrator *deterministically proceeds to a build action each round*; the mathematical treatment that follows is retained for exposition.

3.2.1 Mathematical Formulation of Boids Rules

Let the set of possible actions for an agent be $\mathcal{A}$, which includes building tools of various types ($a_{\text{build},t}$) and using existing tools (a_{use}). Each boids rule produces a preference function $P(\cdot)$ over this action space. In our experiments, $P(\cdot)$ serves as a descriptive scaffold.

3.2.2 Separation: Functional Niche Specialization

The separation rule enforces functional diversity and encourages niche specialization by discouraging the creation of tools that are redundant within an agent's local neighborhood. We model this through two distinct mechanisms.

Saturation-Based Model. This model calculates the saturation $S(t)$ of a given tool type t within the recent history of an agent i's neighborhood $\mathcal{N}_i$. Let $T_{j,\text{recent}}$ be the set of recently created tools by a neighbor j. The saturation is:

$$S(t) = \sum_{j \in \mathcal{N}_i} |\{\tau \in T_{j,\text{recent}} \mid \text{type}(\tau) = t\}|. \tag{1}$$

The preference for building a tool of type t, $P_{\text{sep}}(a_{\text{build},t})$, is modulated by a penalty function $f_{\text{sep}}(S(t))$ that decreases preference as saturation increases:

$$P_{\text{sep}}(a_{\text{build},t}) \propto f_{\text{sep}}(S(t)) = \begin{cases} 0.1 & \text{if } S(t) \geq 2, \\ 0.5 & \text{if } S(t) = 1, \\ 1.0 & \text{if } S(t) = 0. \end{cases} \tag{2}$$

Operationally, saturation counts are gathered over a *recent-neighborhood window* of the last three rounds (default W=3) using *inferred* tool types; the resulting tiers $\{1.0, 0.5, 0.1\}$ are *presented as textual guidance in the prompt*.

Semantic Similarity Model. For a more nuanced differentiation, this model leverages natural language processing. Each tool τ is represented by a TF–IDF vector $\mathbf{v}(\tau)$ derived from its name and functional description. The semantic similarity between a proposed tool τ_p and an existing tool τ_e is their cosine similarity:

$$\text{sim}(\tau_p, \tau_e) = \frac{\mathbf{v}(\tau_p) \cdot \mathbf{v}(\tau_e)}{\|\mathbf{v}(\tau_p)\| \, \|\mathbf{v}(\tau_e)\|}. \tag{3}$$

Conceptually, separation prefers proposals that diverge from nearby artifacts. In practice, the prompt *highlights at most the top two neighbors* whose similarity exceeds a threshold θ=0.30 and includes brief code snippets as "diverge-from" anchors; when there are fewer than two neighbor tools, when vectorization fails, or when both similarity and saturation signals are empty, the rule emits no fragment.

3.2.3 Alignment: Propagation of Successful Strategies

The alignment rule facilitates the propagation of effective behaviors by encouraging agents to learn from their most productive neighbors. Success of a neighbor agent j relative to the current agent i is defined by a productivity function, IsSuccessful(j, i), where success is correlated with the number of tools created ($|T_j| > |T_i|$):

$$\text{IsSuccessful}(j, i) = \begin{cases} 1 & \text{if } |T_j| > |T_i|, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Let $\mathcal{A}_{j,\text{recent}}$ be the set of recent actions performed by agent j. The alignment preference for a, $P_{\text{align}}(a)$, is increased if a has been recently taken by successful neighbors:

$$P_{\text{align}}(a) = P_{\text{base}}(a) + \Delta P_{\text{align}} \cdot \max_{j \in \mathcal{N}_i} \left(\text{IsSuccessful}(j, i) \cdot \mathbb{I}(a \in \mathcal{A}_{j,\text{recent}}) \right), \tag{5}$$

where $\mathbb{I}(\cdot)$ is the indicator function. In the implementation, alignment *filters neighbors to those currently out-producing the focal agent* when such counts are available, and surfaces *narrative exemplars*—complexity, quality, and adoption leaders—within the prompt; if no qualifying exemplars exist, the rule emits no fragment.

3.2.4 Cohesion: Fostering Collaborative Tool Use

The cohesion rule promotes the development of an integrated tool ecosystem by incentivizing agents to use and build upon their neighbors' existing tools. The preference for using tools, $P_{\text{coh}}(a_{\text{use}})$, is

conditioned on the availability of tools in the local environment. Let $N_T = \sum_{j \in \mathcal{N}_i} |T_j|$ be the total number of tools held by all neighbors. The cohesion preference is formulated as:

$$P_{\text{coh}}(a_{\text{use}}) \propto 1 + \delta_{\text{use}} \cdot \mathbb{I}(N_T > 0), \tag{6}$$

where δ_{use} amplifies usage when a local ecosystem exists; a smaller boost δ_{build} encourages complementary builds. In practice, cohesion relies on *recent-neighborhood windows* when available; when no global summary is present, no cohesion fragment is emitted; and fixed coefficients δ_{use}=0.3 and δ_{build}=0.1 are stated directly in the prompt (and accompanying metadata) as communicative guidance.

3.3 Evolutionary Algorithm Module

Evolutionary pressure is introduced through periodic selection and reproduction. Every few rounds, agents are ranked by their average Tool Complexity Index (TCI); the bottom segment is eliminated subject to a minimum-population constraint and replaced via prompt-level crossover or mutation of surviving specializations. This mechanism provides a Darwinian loop in which strategies associated with more complex, reusable tools persist, while less helpful behaviors fade. We study boids-only, evolution-only, and combined conditions against a no-constraint control to isolate contributions of local coordination and global selection. In the present setup, neighborhood structure uses a fixed ring topology, and selection is TCI-based without directly weighting correctness or collaboration metrics.

System performance is evaluated using both correctness and complexity metrics. The TCI measures tool sophistication along code structure, interface design, and compositional reuse. Higher-level indicators capture emergent phenomena such as diversity, specialization divergence, collaboration events, and ecosystem coherence. Experiments are conducted under the fixed neighborhood topology described above; randomized multi-topology replications are left to future work.

4 Experiments & Results

4.1 Tool Complexity Index (TCI)

$$\text{TCI} = \underbrace{C_{\text{code}}}_{[0,3]} + \underbrace{C_{\text{iface}}}_{[0,2]} + \underbrace{C_{\text{comp}}}_{[0,5]} .$$

where $C_{\text{code}} \in [0, 3]$ quantifies code surface, $C_{\text{iface}} \in [0, 2]$ quantifies caller-facing interface burden, and $C_{\text{comp}} \in [0, 5]$ quantifies compositional breadth. All quantities are obtained via static analysis of the tool's `execute` entrypoint and its module directory, without executing code.

Code complexity. We map code surface to a capped linear score $C_{\text{code}} = 3 \min(1, \text{LOC}/300)$, where *LOC* counts effective lines aggregated over the tool directory (excluding blank/comment-only lines). This reflects reading and change costs while preventing size-only inflation via saturation at 300 lines.

Interface complexity. We combine input arity and output surface using $C_{\text{iface}} = \min(1,\, p/5) + \min(1,\, r/5)$, where p is the number of formal parameters of `execute` and the return proxy is $r = \min(5,\, K + D + T)$. Here K is the average top-level key count across dictionary-literal `return` sites, D is the maximum literal nesting depth, and T is top-level kind heterogeneity (number of distinct top-level kinds minus one).

Compositional complexity. We reward modular orchestration using $C_{\text{comp}} = \min(4,\, 0.5\, t) + \min(1,\, 0.1\, e)$, where t counts distinct tools referenced and e counts distinct non-standard-library imports at top level. Prioritizing breadth over depth encourages decomposition into reusable components while acknowledging ecosystem surface without letting external dependencies dominate.

Parsing quality gate. A parsing quality gate down-weights the raw sum by a factor of 0.6 when the module fails to parse to an AST, ensuring syntactically invalid tools are retained for analysis but penalized in ranking.

Table 1: Baseline (Global) vs Boids+Evolution per task. Code/Comp. are last-round complexities.

Metric	Text Analysis		Data Science		Simulation/Modeling	
	Baseline	Boids	Baseline	Boids	Baseline	Boids
Tools created	9	13	10	18	14	16
Pass rate (%)	89	85	100	89	86	88
Avg. TCI	2.10	1.95	2.45	2.32	2.07	2.05
Code complexity	0.50	0.52	0.86	0.88	0.63	0.64
Compos. complexity	0.21	0.03	0.19	0.04	0.03	0.01

4.2 Observations

Evolutionary-Boids reliably lifts throughput and balances contribution across agents without materially harming pass rates, but under current prompting it suppresses multi–tool composition.

Across all three tasks, *Evolutionary-Boids* produces more artifacts per run under identical agent counts and rounds (*13–18* vs. *9–14* in the global baselines; cf. Table 1). The gain is ecosystem-wide rather than star-driven: the most productive agent accounts for a smaller share of total output (*lower top_share*), and the leading contributors each deliver *3–5* tools rather than a single dominant performer. Reliability remains comparable overall (mid– to high–80% pass rate), with one global data-science run reaching *100%* on fewer attempts. In contrast, Boids increases the number of attempts ("shots on goal") without noticeably degrading correctness, suggesting that lightweight, failure-aware retries could close the residual gap while preserving the throughput advantage.

Complexity differences manifest primarily in composition rather than unit difficulty. The average tool complexity index (TCI) remains in the *2.0–2.5* band for all regimes, indicating similar per-artifact difficulty. However, *compositional* complexity is near-zero under Boids, while global baselines trend toward $\sim$0.2 due to more frequent reuse/chaining of prior tools. Flocking cues (alignment/separation) and per-round build enforcement stimulate fresh tool ideation, yet the current prompts—and a conservative dependency policy—bias agents toward self-contained solutions. Reflection traces corroborate this: agents consistently propose distinct primitives even after failures but rarely invoke existing ones.

Synthesis. Stepping back, our findings suggest a design pattern for multi-agent tool ecosystems: begin with *breadth-first modular exploration* driven by simple local rules, then transition—under explicit incentives and constraints—toward *depth-oriented assembly* and integrated pipelines. In this view, evolution functions as institutional governance: selection retains artifacts and interaction rules that are socially useful, gradually specializing the ecosystem while preserving optionality. The practical guidance is to couple diversity pressure (separation, novelty, quality–diversity) with time-varying mechanisms that nudge reuse (composition quotas, reward shaping, lineage-aware telemetry), yielding heavier but more integrated workflows only when the parts bin is sufficiently rich. Beyond engineering benefits (maintainability, reuse, fault isolation), this staged strategy offers a tractable scientific handle on emergent intelligence: it externalizes coordination into measurable institutions and lets *usefulness* act as the unifying currency across agents, artifacts, and generations.

5 Conclusions and Limitations

5.1 Conclusions

In this paper, we introduced *Evolutionary-Boids*, a simple yet effective coordination mechanism for large-scale multi-agent tool generation. By combining flocking dynamics with evolutionary specialization, the framework transforms local behavioral rules into emergent collective productivity: agents self-organize to explore diverse regions of the design space while maintaining steady reliability and balance. Empirically, Evolutionary-Boids produced 30–50% more valid artifacts than global baselines across three domains, distributing output more evenly and expanding the system's functional coverage without compromising pass rates.

However, our analysis also revealed a clear trade-off between *breadth* and *depth*: while the method excels at generating a broad range of primitives, it falls short in spontaneous tool reuse and multi-step composition. This finding frames Evolutionary-Boids as an *early-phase engine* for rapid ideation—building the component library upon which deeper coordination can later emerge. Looking ahead, we envision extending the framework to support composition-aware prompting, adaptive reward shaping, and evolutionary selection that favors cooperative behaviors. Beyond improving technical metrics, these extensions will allow us to study how simple coordination rules scale into structured, self-improving ecosystems—an essential step toward understanding collective intelligence in open-ended agent systems.

Evolutionary-Boids is well-suited to the *early phase* of system construction: it rapidly populates a diverse "parts bin" while maintaining broad participation across the agent pool. For downstream stages that prioritize multi-step pipelines and reuse of shared utilities, the framework should make reuse both *salient* and *rewarded*. Concretely, we recommend (i) generation-time salience and guards (memory traces that surface relevant prior tools; AST checks that reject self-contained proposals when feasible reuse exists), (ii) evaluation-time incentives (rubric bonuses and pass criteria that require at least one correct prior-tool invocation where applicable), and (iii) failure-aware retries targeted at composed artifacts. A sandboxed whitelist of third-party dependencies (e.g., `numpy`, `pandas`) can safely raise the ceiling on legitimate composition. We will track deltas in pass rate, TCI and more to quantify the effect of these interventions.

5.2 Limitations

This study is a preliminary step toward answering a broader question: how far can flocking-style coordination and lightweight evolution push multi-agent tool building before explicit planning and strong composition constraints become necessary? While our results are encouraging, they should be interpreted with care.

Our prompts currently *encourage* but do not *enforce* reuse, which likely contributes to low compositional complexity under Boids. Execution-safety restrictions on third-party libraries may further inhibit legitimate chaining. Evolutionary pressure was not fully exercised in the reported sessions (no pruning occurred), limiting conclusions about selection dynamics. Finally, the tasks studied (text analysis, data science, simulation) do not cover domains where long-horizon composition is intrinsic (e.g., robotics, multimodal research pipelines). Future work will harden composition constraints and rewards, introduce a sandboxed dependency whitelist, scale population size/rounds and selection intensity to realize prune/replicate/mutate dynamics, and evaluate in settings that demand chaining to test whether Evolutionary-Boids can transition from breadth-first exploration to depth-oriented assembly.

Our findings reflect the current instrumentation and scale. *Construct validity:* TCI-Lite is a static proxy and does not capture runtime semantics, side-effects, or true data-flow depth; compositional complexity may undercount "soft reuse" (e.g., protocol alignment without explicit calls). *Internal validity:* telemetry is partial (limited adoption graphs, shallow call-graph instrumentation), and the sandbox may mask latency/cost trade-offs or swallow failure modes that matter in production. *External validity:* experiments are Python-centric with small populations, short horizons, and few seeds; tests emphasize internal consistency over downstream task utility, and dependency policies restrict otherwise reasonable compositions. To mitigate these issues we matched budgets/prompts across regimes and release structured logs plus per-run artifacts; the sandbox is designed to extend with rule-level telemetry and runtime metrics (adoption graphs, call-graph depth, latency/cost) to strengthen attribution and connect modularity gains to end-task usefulness.

Two additional threats merit emphasis. *Prompt/policy confounds:* the observed throughput gains and lower composition could be driven as much by per-round build enforcement and "novelty-seeking" cues as by flocking itself; targeted ablations (flocking only, enforcement only, retry only) are needed to disentangle contributions. *Stochasticity and reproducibility:* with limited seeds and unswept hyperparameters, variance remains; future releases will pin toolchains, containerize environments, and report confidence intervals to bound effect sizes.

In short, our current framework prioritizes *breadth*—rapidly populating a diverse parts bin—over *depth*. This is a deliberate early-phase choice, not a fundamental limitation of the paradigm. The next iteration will explicitly test the transition from exploration to assembly via composition quotas,

reward shaping, dependency whitelists, richer telemetry, larger/evolving populations, and evaluation on tasks where chaining is the dominant success mode.

Acknowledgements

This work is an experiment in conducting research with AI. The authors aim to use AI systems to explore an exciting question in artificial intelligence—how such systems can scale effectively. Scalability remains an open challenge for large-scale automation, particularly for LLM-based agents, and we believe that simple local rules may offer an inspiring path forward. We gratefully acknowledge the generous support of Pulse Lab (SysteMind) and thank the organising committee for creating a uniquely AI-led conference and for the opportunity to present this work.

For replication, we instantiate the `AzureOpenAIClient`, which retrieves the Azure endpoint, API key, and the `gpt-4.1-nano` deployment (API version `2024-12-01-preview`) from the environment template, along with default temperature constants of 0.7 for free-form calls and 0.1 for structured outputs. The client remains fixed to this deployment when issuing chat completions with a 30,000-token ceiling.

The agent loop then defines a stage-specific sampling schedule: reflections query the model at temperature 0.7; tool blueprints reduce this to 0.5 for more deterministic specifications; concrete tool code generation lowers it further to 0.1 to minimize drift; and the subsequent unit tests also execute at 0.1 under the same model.

References

[1] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH '87*, pages 25–34. ACM, 1987.

[2] Tamás Vicsek, András Czirók, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75(6):1226–1229, 1995.

[3] John Toner and Yuhai Tu. Long-range order in a two-dimensional dynamical model: How birds fly together. *Physical Review Letters*, 75(23):4326–4329, 1995.

[4] David J. T. Sumpter. The principles of collective animal behaviour. *Philosophical Transactions of the Royal Society B*, 361(1465):5–22, 2006.

[5] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282–4286, 1995.

[6] Reza Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

[7] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.

[8] Iain D. Couzin, Jens Krause, Nigel R. Franks, and Simon A. Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025):513–516, 2005.

[9] Thomas S. Ray. An approach to the synthesis of life. In Christopher G. Langton, Charles Taylor, J. Doyne Farmer, and Steen Rasmussen, editors, *Artificial Life II*, volume XI of *Santa Fe Institute Studies in the Sciences of Complexity*, pages 371–408. Addison-Wesley, Redwood City, CA, 1991.

[10] Charles Ofria and Claus O. Wilke. Avida: A software platform for research in computational evolutionary biology. *Artificial Life*, 10(2):191–229, 2004.

[11] Joel Lehman and Kenneth O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.

[12] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. arXiv:1504.04909, 2015.

[13] Matthew C. Fontaine, Julian Togelius, Stefanos Nikolaidis, and Amy K. Hoover. Covariance matrix adaptation for the rapid illumination of behavior space. arXiv:1912.02400, 2020.

[14] Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O. Stanley. Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions, 2019.

[15] Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula, 2019.

[16] Micah Carroll, Rohin Shah, Mark K. Ho, Thomas L. Griffiths, Sanjit A. Seshia, Pieter Abbeel, and Anca D. Dragan. On the utility of learning about humans for human-ai coordination. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[17] Open Ended Learning Team, Adam Stooke, Anuj Mahajan, et al. Open-ended learning leads to generally capable agents. arXiv:2107.12808, 2021.

[18] Joseph Suárez, Phillip Isola, Kyoung Whan Choe, et al. Neural mmo 2.0: A massively multi-task addition to massively multi-agent learning. In *NeurIPS Datasets & Benchmarks*, 2023.

[19] Timo Schick, Jane Dwivedi-Yu, et al. Toolformer: Language models can teach themselves to use tools. arXiv:2302.04761, 2023.

[20] Guanzhi Wang et al. Voyager: An open-ended embodied agent with large language models. arXiv:2305.16291, 2023.

[21] Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for "mind" exploration of large language model society, 2023.

[22] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation, 2023.

[23] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. arXiv:2304.03442, 2023.

[24] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.

[25] Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. In *NeurIPS*, volume 33, pages 13049–13061, 2020.

[26] John P. Agapiou, Alexander Sasha Vezhnevets, Edgar A. Duéñez-Guzmán, et al. Melting pot 2.0, 2022.

[27] Cornelius Ruhdorfer et al. The overcooked generalisation challenge. arXiv:2406.17949, 2024.

[28] Da Yin, Faeze Brahman, Abhilasha Ravichander, Khyathi Chandu, Kai-Wei Chang, Yejin Choi, and Bill Yuchen Lin. Agent lumos: Unified and modular training for open-source language agents. In *ACL*, pages 12380–12403, 2024.

[29] Ziyi Yang, Zaibin Zhang, Zirui Zheng, Yuxian Jiang, Ziyue Gan, Zhiyu Wang, Zijian Ling, Jinsong Chen, Martz Ma, Bowen Dong, et al. Oasis: Open agent social interaction simulations with one million agents, 2024.

[30] Mengkang Hu, Yuhang Zhou, Wendong Fan, Yuzhou Nie, Bowei Xia, Tao Sun, Ziyu Ye, Zhaoxuan Jin, Yingru Li, Qiguang Chen, et al. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation, 2025.

[31] Sha Li, Revanth Gangi Reddy, Khanh Duy Nguyen, Qingyun Wang, Yi (May) Fung, Chi Han, Jiawei Han, Kartik Natarajan, Clare R. Voss, and Heng Ji. Schema-guided culture-aware complex event simulation with multi-agent role-play. In *EMNLP System Demonstrations*, pages 372–381, 2024.

[32] Zhitao He, Zijun Liu, Peng Li, Yi R. Fung, Ming Yan, Ji Zhang, Fei Huang, and Yang Liu. Advancing language multi-agent learning with credit re-assignment for interactive environment generalization, 2025.

Agents4Science AI Involvement Checklist

1. **Hypothesis development**: Hypothesis development includes the process by which you came to explore this research topic and research question. This can involve the background research performed by either researchers or by AI. This can also involve whether the idea was proposed by researchers or by AI.

   Answer: **[C]**

   Explanation: The problem framing and hypothesis were initiated by the authors but developed and iterated by AI: unify Boids-style local rules with evolutionary selection to study tool-creating agent societies, instrumented via a measurement-first sandbox and TCI.

2. **Experimental design and implementation**: This category includes design of experiments that are used to test the hypotheses, coding and implementation of computational methods, and the execution of these experiments.

   Answer: **[C]**

   Explanation: Humans aid in establishing the overall framework and protocol guidance. In this context, AI agents are responsible for designing the experiment, coding, and executing the experiments under human supervision.

3. **Analysis of data and interpretation of results**: This category encompasses any process to organize and process data for the experiments in the paper. It also includes interpretations of the results of the study.

   Answer: **[D]**

   Explanation: From runs produced structured logs/JSON, the aggregation, comparison (Boids vs. baseline), and interpretation (e.g., modularity signature, redundancy/diversity trends) were all performed by AI (§§4.2–4.3; §5).

4. **Writing**: This includes any processes for compiling results, methods, etc. into the final paper form. This can involve not only writing of the main text but also figure-making, improving layout of the manuscript, and formulation of narrative.

   Answer: **[D]**

   Explanation: The manuscript's text, figure/table selection, and narrative (methods, results, limitations, broader impacts) were mainly written by LLM. Human is involved for supervising this process.

5. **Observed AI Limitations**: What limitations have you found when using AI as a partner or lead author?

   Description: AI still has context window limitations, which means that in long conversations, it is hard to follow instructions. In addition, AI tends to overcomplicate things and adds too many details at once.

Agents4Science Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper claims a unified Boids and evolution framework and reports tool-cereating outcomes across domains; Methods, Experiments and Conclusions section support these claims within scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In conlusions and limitations section the paper details construct/internal/external/conclusion validity limits.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The work formalizes decision rules/metrics but does not present theorems; no proofs are required.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The agent loop, defaults, observables, and protocols are specified; runs emit structured logs or JSON with per-run directories for rebuilding tables/figures.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The submission provides logs, JSON traces, run folders and github repo and explains how to reconstruct results from these artifacts.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the Agents4Science code and data submission guidelines on the conference website for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: It reports agent counts/rounds, neighborhood size, separation threshold, decision weights, TCI formula, and evaluation metrics.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper describes replicated runs across randomized initializations/topologies.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, or overall run with given experimental conditions).

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [NA]

   Justification: API used for each experiment.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the Agents4Science Code of Ethics (see conference website)?

   Answer: [Yes]

   Justification: No human subjects or sensitive data.

   Guidelines:

   - The answer NA means that the authors have not reviewed the Agents4Science Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Positive implications for alignment/governance and evolving tool ecosystems are discussed alongside risks.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations, privacy considerations, and security considerations.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies.