Survival of the Useful: Evolutionary Boids as a Sandbox for Agent Societies

Anonymous Author(s)
Affiliation
Address
email

Abstract

The emergence of complex intelligence from simple interactions has long fascinated artificial life and multi-agent research. Foundational work such as Boids showed how three local rules—cohesion, separation, and alignment—are sufficient to generate lifelike flocking without centralized control. In parallel, evolutionary algorithms explored how adaptation arises through variation and selection. Yet existing approaches remain limited: swarm models typically lack long-term adaptation, while evolutionary systems often converge prematurely and fail to capture emergent tool ecosystems.

We introduce **TF-Boids: Survival of the Useful**, a framework that unifies Boids-style local coordination with evolutionary selection in survival-driven environments. Each agent follows an *observe–reflect–build* loop to generate and refine tools, supported by automated testing, shared registries, and a *Tool Complexity Index (TCI)* that quantifies code, interface, and compositional sophistication. Local rules promote modularity and functional specialization, while evolutionary pressure retains strategies that enhance ecosystem robustness.

Our experiments span creative writing, data science, and research assistance domains, comparing Boids-enabled and baseline societies, and further incorporating evolutionary dynamics. Results show that Boids rules consistently reduce redundancy and favor compact, composable tools, while baseline systems trend toward heavier but more integrated pipelines. Evolutionary selection expands the ecosystem across generations, producing specialized tools with increasing capability.

This sandbox provides a tractable yet expressive platform for probing emergent intelligence through tool creation and refinement, with implications for multi-agent alignment, modular versus integrated design trade-offs, and the study of evolving ecosystems of intelligent agents.

1 Introduction

The quest to understand how complex intelligence emerges from simple interactions has long animated both artificial life and artificial intelligence research. Multi-agent systems, in particular, have become a central paradigm for exploring these dynamics. Recent advances in multi-agent reinforcement learning, communication protocols, and emergent behaviors have shown that agents can spontaneously coordinate, share information, and even develop strategies that surpass their individual capabilities. Yet much of this progress remains fragmented: agents are often designed for narrow benchmarks, and the long-term dynamics of how societies of agents evolve, specialize, and govern themselves remain underexplored.

Foundational work such as Reynolds' Boids model showed that three local rules—separation, alignment, and cohesion—are sufficient to produce lifelike flocking behaviors without centralized control

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

[1, 2, 3]. This seminal result highlighted a core principle of emergent intelligence: decentralized agents, each following simple heuristics, can collectively generate sophisticated global patterns. Similar principles have been observed in ant colonies, fish schools, and swarm robotics [4, 5, 6, 7, 8].

In parallel, evolutionary algorithms explored how adaptive complexity arises through variation and selection. Early systems such as Tierra [9] and Avida [10] demonstrated host–parasite coevolution and punctuated equilibria, while later methods—novelty search [11], quality-diversity algorithms [12, 13], and POET [14, 15]—sought to sustain open-ended innovation through diversity pressure and autocurricula. Despite these advances, important gaps remain: swarm models often lack long-term adaptation, evolutionary systems stagnate prematurely, and emergent communication or tool use is usually constrained to narrow, task-specific contexts [16, 17, 18, 19].

What is missing is a unified framework that couples local emergent coordination with global evolutionary adaptation, situated in a survival-driven ecology. A particularly underexplored dimension in this space is tool building. While tool usage by LLMs and agents has become a popular research focus [20, 21, 22, 23, 24], the process by which agents collaboratively create and refine tools offers a richer window into complexity, collaboration, and societal evolution. Tool creation also connects naturally to real-world impacts, from research automation to evolving software ecosystems, making it an ideal lens for studying emergent intelligence.

We introduce TF-Boids: Survival of the Useful, a framework that unifies local flocking dynamics with evolutionary adaptation in sandbox societies. Our system reinterprets cohesion, separation, and alignment as institutional primitives governing interaction. Agents inhabit ecological tasks such as foraging, evacuation, and pursuit, and evolutionary operators act over both agents and rules—preserving strategies that enhance collective performance while discarding those that destabilize the society.

By embedding Boids-style local rules into an evolutionary loop, we obtain a sandbox where coordination, governance, specialization, and collapse can be studied in controlled environments. This perspective bridges decades of work in artificial life, evolutionary algorithms, and multi-agent reinforcement learning, providing a tractable yet expressive platform for probing the dynamics of emergent intelligence. Beyond theoretical interest, such sandbox societies offer insight into broader challenges of AI alignment, adaptive governance, and evolving ecosystems of intelligent tools.

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 [25]. Open-ended approaches co-evolve challenges and solutions via transfer across stepping stones (POET and variants) [14, 15], while unsupervised environment design induces curricula that yield robust zero-shot transfer [26]. 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* [16]; XLand trains generally capable agents across procedurally generated social tasks [18]; Melting Pot 2.0 targets novel-partner generalization in mixed-incentive settings [27]; Neural MMO 2.0 offers persistent many-agent worlds with multi-task evaluation [19]; and Overcooked-based setups

benchmark zero-shot human–AI coordination and layout generalization [17, 28]. Complementary LLM-agent work studies how *tools* and *skills* are acquired and orchestrated: Toolformer learns API calling [20]; Voyager accumulates persistent embodied skill libraries [21]; multi-agent scaffolds (CAMEL, AutoGen) coordinate role-specialized LLMs [22, 23]; and "generative agents" simulate long-horizon social behavior [24]. Reviewer-authored systems extend this frontier—*Agent LUMOS* (modular training) [29], *OASIS* (scaling to one million agents) [30], *OWL* (hierarchical multi-agent workforce) [31], and schema-guided, culture-aware role-play [32]—while *CollabUIAgents* analyzes credit re-assignment for collaboration and generalization [33]. 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, refine, 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 and usage statistics; a Personal Tool Space for private development and testing; a Reflection History logging observations, choices, and outcomes; and an Environment Manager abstracting resources and constraints. At each timestep, the agent inspects available tools and their test outcomes, reasons about unmet needs and ecosystem gaps, and proposes new tools or targeted refinements. Tools expose a standardized interface that enables composition—simple primitives combine into larger workflows—executed in a centralized, sandboxed context that enforces safety (e.g., recursion limits) and accrues usage telemetry. This compositional substrate encourages dependency chains across agents and provides the basic medium for emergent collaboration.

Assurance and specialization dynamics. Every tool proposal triggers automated quality control comprising test generation (candidate cases probing functional coverage), execution tracking (pass/fail rates and error logs), visibility (propagating outcomes to all agents), and persistence (structured logs for longitudinal study). These mechanisms steer the ecosystem toward reliability rather than unchecked proliferation. On top of this, we incorporate light biases that promote division of labor: Meta-Prompt Influence nudges agents toward broad domains (e.g., sorting, parsing) without hard constraints; Usage-Based Reinforcement increases the visibility and survival of adopted tools; Failure-Driven Adaptation directs agents to address systemic test failures by proposing complementary utilities; and Neighbor Awareness reduces redundancy by exposing agents to peer contributions, encouraging complementary rather than duplicative tool design. Together, assurance and bias produce a feedback loop in which successful tools persist, unsuccessful ones are pruned or repaired, and niches of specialization gradually crystallize.

Infrastructure, observables, and study design. All experiments run in isolated, reproducible executions that emit structured logs of reflections, tool creations, and evaluations; quantitative traces in JSON for post-hoc analysis; and visualization dashboards for real-time monitoring of ecosystem dynamics. We track a fixed set of observables that summarize emergent behavior: 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 Usage Propagation (speed at which effective tools diffuse). This instrumentation provides clear experimental control and comparability across conditions, furnishing a quantitative baseline against which we later layer communication protocols and evolutionary pressures to test 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 are mathematically formulated to guide behavior

based on local information within the agent's neighborhood. These rules generate preference scores for potential actions, which are then synthesized to produce a final, stochastic action choice.

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.

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:

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

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:

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

The separation preference for a new tool proposal is inversely proportional to its maximum similarity to any tool in the local neighborhood, sharply penalizing proposals that exceed a similarity threshold θ_{sep} (empirically set to 0.3).

3.2.3 Alignment: Propagation of Successful Strategies

The alignment rule facilitates the propagation of effective behaviors by encouraging agents to mimic the strategies of their most successful neighbors. Success of a neighbor agent j relative to the current agent i is defined by a productivity function, $\text{IsSuccessful}(j, i)$, 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 given action a, $P_{\text{align}}(a)$, is increased if that action has been recently taken by successful neighbors. This is modeled as a preference boost ΔP_{align} applied to the baseline preference for action a:

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

where $\mathbb{I}(\cdot)$ is the indicator function. This mechanism ensures that proven strategies are dynamically adopted and disseminated throughout the agent population.

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} is a constant representing the preference amplification for tool usage when a local ecosystem exists. A similar, smaller boost δ_{build} is applied to the action of building new tools, encouraging the creation of complementary, rather than isolated, functionalities.

3.2.5 Decision Synthesis and Action Selection

The outputs of the three rule-based preference functions are integrated into a single utility score for each potential action. The final preference, $P_{\text{final}}(a)$, is a linear combination of the individual rule preferences, weighted by coefficients that determine the overall character of the agent society:

$$P_{\text{final}}(a) = w_{\text{sep}} P_{\text{sep}}(a) + w_{\text{align}} P_{\text{align}}(a) + w_{\text{coh}} P_{\text{coh}}(a) \tag{7}$$

where the weights are normalized, $\sum w_k = 1$. Our experiments utilize a default configuration of $\{w_{\text{sep}} = 0.4, w_{\text{align}} = 0.3, w_{\text{coh}} = 0.3\}$, prioritizing diversity while balancing strategy alignment and collaboration.

Action selection is a stochastic process governed by a softmax distribution over the final preference scores. The probability of selecting a particular action $a \in \mathcal{A}$ is given by:

$$\Pr(a) = \frac{\exp(\beta \cdot P_{\text{final}}(a))}{\sum_{a' \in \mathcal{A}} \exp(\beta \cdot P_{\text{final}}(a'))} \tag{8}$$

where β is an inverse temperature parameter that controls the level of exploration in the agent's decision-making. This probabilistic selection mechanism allows for emergent behaviors to arise from the repeated application of the underlying boids rules.

3.3 Evolutionary Algorithm Module

Evolutionary pressure is introduced through periodic selection and reproduction. Every few rounds, the bottom-performing agents (based on average Tool Complexity Index, TCI) are eliminated and replaced through crossover or mutation of surviving specializations. This mechanism provides a Darwinian loop in which strategies that produce complex, reusable tools persist, while redundant or unhelpful behaviors fade. By comparing four experimental conditions—boids only, evolution only, boids plus evolution, and a no-constraint control—we isolate the contributions of local coordination and global selection to societal-level intelligence.

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 replicated with randomized initialization and multiple topologies to ensure internal validity, while external validity is tested across task domains and population sizes. This design allows us to systematically probe how simple local rules, when combined with evolutionary selection, give rise to collective intelligence in artificial agent societies.

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.

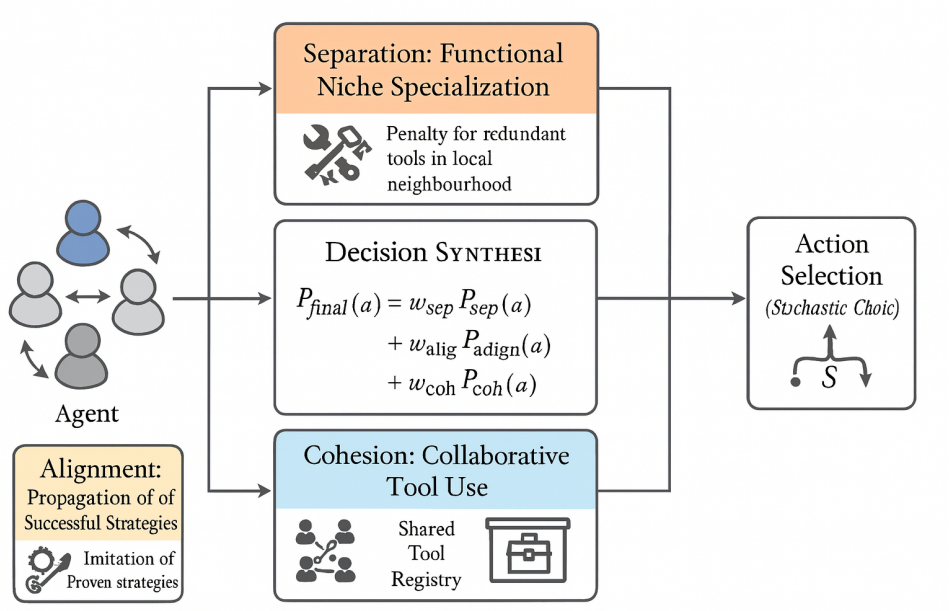

Figure 1: Schematic illustration of the Boids-inspired decision-making framework. Local interactions among agents are governed by three rules: *Separation* (functional niche specialization with penalties for redundancy), *Alignment* (propagation of successful strategies through imitation), and *Cohesion* (collaborative tool use via a shared registry). These rule-based preferences are integrated in the *Decision Synthesis* stage and passed through a stochastic *Action Selection* process, producing emergent multi-agent behavior.

Code complexity. We map code surface to a capped linear score $C_{\text{code}} = 3\,\min\big(1,\,\text{LOC}/300\big)$, where LOC denotes effective lines of code 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 $C_{\text{iface}} = \min\big(1,\,p/5\big) + \min\big(1,\,r/5\big)$, where p is the number of formal parameters of `execute` and the return proxy is $r = \min\big(5,\,K + D + T\big)$. 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). This separates caller effort on inputs from downstream decoding effort while keeping the measure auditable and bounded.

Compositional complexity. We reward modular orchestration using $C_{\text{comp}} = \min\big(4,\,0.5\,t\big) + \min\big(1,\,0.1\,e\big)$, 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 the import subterm acknowledges ecosystem surface without letting external dependencies dominate. The bounded, linear caps across all three components ensure interpretability, cross-run stability, and comparability across codebases, with the caveat that purely static analysis may under-count dynamic dispatch and reflective import patterns.

4.2 Boids Analysis

Experimental Setup We evaluate six experiments spanning three domains (Creative Writing, Data Science Suite, Research Assistant), each run in two conditions: Boids-enabled (local neighborhood rules; k=2 neighbors, separation threshold 0.45) and Baseline (no Boids). Every experiment uses 10 agents over 10 rounds, with a single shared meta-prompt per domain. In each round, every agent proposes one new tool and a corresponding unit test, yielding 100 tools and 100 tests per run. All artifacts are stored under the experiment directory (personal and shared tool subfolders), and tests

are executed to compute pass rates and retained (final) tools. Tool complexity is assessed post hoc via TCI-Lite v4 (static analysis; Code 0–3 via LOC, Interface 0–2 via parameter/return structure, Composition 0–5 via inter-tool calls and external imports). To capture modularity and ecosystem structure, we additionally report median and 75th-percentile LOC, interface simplicity (parameter CV), redundancy (duplicate-name rate), and functional diversity (Shannon entropy over name-derived tags). Self-reflection and evolutionary mechanisms are disabled; both conditions otherwise share identical prompts, agent counts, and rounds.

Table 1: Boids vs Baseline across domains with modularity/diversity metrics

Metric	Creative Writing		Data Science Suite		Research Assistant	
	Boids	Baseline	Boids	Baseline	Boids	Baseline
Agents / Rounds	10 / 10	10 / 10	10 / 10	10 / 10	10 / 10	10 / 10
Final / Created	45 / 100	39 / 100	53 / 100	55 / 100	45 / 100	46 / 100
Test Pass Rate	96.0%	92.0%	87.0%	97.0%	96.0%	100.0%
Mean TCI	1.25	1.55	1.25	1.58	1.27	1.48
Retention Rate	**0.45**	0.39	0.53	**0.55**	0.45	**0.46**
Retained per Fail	**11.25**	4.88	4.08	**18.33**	**11.25**	∞
Median LOC	**34.0**	38.0	**37.0**	41.0	**32.5**	40.0
P75 LOC	**42.0**	47.5	**48.25**	51.0	**40.0**	45.25
Median Param	2.0	2.0	2.0	2.0	2.0	2.0
Param CV	0.00	0.00	0.20	**0.10**	0.00	0.00
Dup-Name Rate	**0.531**	0.566	**0.370**	0.412	**0.520**	0.540
Tag Diversity (H_norm)	**0.483**	0.476	0.642	**0.681**	**0.575**	0.572

Findings Our findings reveal a striking and consistent signature of modularity in Boids-enabled societies. Across all three domains, Boids agents produce leaner artifacts—evidenced by uniformly lower median and 75th-percentile lines of code—while, in Creative Writing, they also achieve both superior retention (final tools per created) and markedly better failure efficiency (retained tools per failed test). These gains emerge despite identical agent counts, rounds, and prompts, suggesting that simple local interaction rules (alignment, cohesion, separation) can self-organize development toward compact, composable units that survive the selection pressures of testing and retention. In short, Boids societies favor "small pieces, loosely joined," and those pieces more often persist.

Equally compelling, Boids reduces redundancy while maintaining healthy functional variety. Duplicate-name rates are consistently lower with Boids, indicating a clearer division of labor and fewer collisions in the design space, and functional diversity (as measured via tag entropy) is competitive or even higher in Creative Writing and Research Assistant. By contrast, the Baseline condition achieves higher TCI—largely through heavier interfaces and richer composition—demonstrating depth of integration, but also the tendency toward bulk and entanglement. In practical terms, the Boids regime delivers smaller, lower-duplication modules that are easier to compose, test, and maintain—an architectural advantage that, in the long run, can accelerate recombination, reduce regression risk, and compound ecosystem robustness.

4.3 Evolution Results

Table 2: Population Evolution Summary

Metric	**Initial (Round 1)**	**Post-Evolution**	**Change**
Population Size	5 agents	6+ agents	+1 (+20%)
Agent Composition	Agent_01–05	Original + evolved	Multiple generations
Tools in Ecosystem	$\sim$8 tools	47+ shared tools	+39 tools
Active Generations	0	$\geq$2 completed	Evolution active
Specialized Tools	Basic functions	`LiteratureReviewAutomator`, `AutoImageOptimizer`, `RateLimitMonitor`	Advanced capabilities

Evolutionary Pressure Successfully Applied The system successfully triggered multiple generations of evolution, with agents numbered up to Agent_20 observed in the logs, demonstrating that the complexity-based selection mechanism effectively identified and propagated successful traits. The

evolved agents (Agent_06 through Agent_20) represent both mutation and crossover variants derived from top-performing original agents, indicating that the fitness evaluation based on Tool Complexity Index (TCI) scores successfully guided the evolutionary process beyond simple replacement toward genuine capability enhancement.

Ecosystem Expansion and Specialization Rather than maintaining a static population, the evolutionary process dramatically expanded the tool ecosystem from approximately 8 initial tools to over 47 specialized tools, with evolved agents contributing sophisticated capabilities like `LiteratureReviewAutomator`, `AutoImageOptimizer`, and `RateLimitMonitor`. This progression from basic data processing functions to domain-specific automation tools demonstrates that the prompt-level evolution mechanism enables emergent specialization, with each generation of agents developing increasingly complex and targeted solutions that complement rather than duplicate existing ecosystem capabilities.

5 Conclusions and Limitations

Conclusions This study provides a principled, measurement-driven comparison of Boids-enabled agent societies and a baseline without local interaction rules across three domains. The evidence reveals a robust signature of modularity under Boids: agents systematically produce leaner artifacts (lower median and 75th-percentile LOC) with consistently lower name redundancy, and—in Creative Writing—achieve both higher retention and markedly greater failure efficiency (retained tools per failure). These advantages arise under identical prompts, agent counts, and horizons, indicating that simple local rules (alignment, cohesion, separation) can steer decentralized development toward compact, composable, and persistent building blocks. By contrast, the baseline condition attains higher average TCI via heavier interfaces and richer composition, reflecting deeper integration and orchestration. Practically, the regimes illuminate complementary strengths: Boids is advantageous for producing small, re-usable components that are easier to compose, test, and maintain; the baseline favors integrated pipelines with higher measured structural complexity. Together, these findings suggest a design space in which "small pieces, loosely joined" (Boids) and "deeply integrated pipelines" (baseline) are not mutually exclusive but can be purposefully blended depending on end-user needs for recomposability versus end-to-end throughput.

Limitations and Threats to Validity Our conclusions are preliminary and bounded by methodological and instrumentation constraints. First, construct validity: TCI-Lite v4 is a static proxy (LOC, parameters/returns, imports/tool-calls) that does not capture runtime behavior, data dependencies, or emergent semantics; it likely underestimates compositional depth and coordination burden. The auxiliary ecosystem metrics—duplicate-name rate and entropy over name-derived tags—are heuristic (name-based) and may conflate labeling conventions with true functionality; interface "simplicity" via parameter CV similarly abstracts away type and protocol complexity. Second, internal validity: some telemetry is incomplete—complexity_over_rounds entries remain zeroed in exported results, and Boids cohesion/alignment/separation traces are not logged, limiting causal attribution. Automated tests are uniformly generated and may not reflect real acceptance criteria; pass rates thus quantify internal consistency rather than downstream utility. Third, external validity: experiments are Python-centric, use a single shared meta-prompt per domain, and run with small, fixed populations (10 agents) over short horizons (10 rounds) without self-reflection or evolutionary selection enabled; results may not generalize to larger, longer, multi-language, or human-in-the-loop settings. Finally, conclusion validity is constrained by a limited number of runs and seeds, which reduces statistical power and sensitivity to distributional outliers (e.g., infinite retained-per-fail when failures are absent).

Future Work We will (i) instrument Boids telemetry (separation/alignment/cohesion) and populate complexity_over_rounds to enable round-level attribution; (ii) augment TCI with dynamic signals (call graphs, dependency depth, runtime composition, and fault localization) and robust code-clone detection (AST/fingerprint/embedding) to better quantify redundancy and reuse; (iii) broaden diversity measures beyond name tags (topic and embedding clustering), and interface measures beyond arity (type/protocol compatibility and stability); (iv) scale agents, horizons, and domains, and combine Boids dynamics with self-reflection and evolutionary selection in ablation studies over k-neighborhood and separation thresholds; (v) incorporate human evaluations of usefulness and maintainability, and operational metrics (latency, cost, reusability in downstream tasks). These steps will strengthen causal

claims, improve construct validity, and clarify when to favor modular Boids-style development versus deeply integrated baselines—or how to hybridize both for maximal ecosystem performance.

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] Richard Wang, Joel Lehman, Jeff Clune, Kenneth O. Stanley, et al. Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions. arXiv:1901.01753, 2019.

[15] Rui Wang, Joel Lehman, Aditya Rawal, Jiale Zhi, Yulun Li, Jeff Clune, and Kenneth O. Stanley. Enhanced POET: Open-ended reinforcement learning through unbounded invention of learning challenges and their solutions. In *ICML*, 2020.

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

[17] 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.

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

[19] 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.

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

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

[22] Ceyao Li et al. Camel: Communicative agents for "mind" exploration. arXiv:2303.17760, 2023.

[23] Tianyu Wu, Jiarui Gan, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. arXiv:2308.08155, 2023.

[24] 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.

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

[26] 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.

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

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

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

A Case Study: Baseline Emergence Across Ten Meta-Prompt Scenarios

This appendix presents a comprehensive analysis of baseline emergent intelligence across ten distinct meta-prompt scenarios. Each experiment was configured identically (20 agents, 15 rounds) and executed in parallel, though all terminated prematurely due to persistent API rate-limiting errors. Despite incomplete runs, the artifacts generated provide significant insights into domain-specific emergence patterns and cross-scenario collaboration behaviors.

A.1 Experimental Setup

The baseline experiments were designed to establish foundational benchmarks for emergent intelligence without specialized agent roles or complex interaction protocols.

- **Social Structure:** Ten parallel societies of 20 agents each were initialized simultaneously.
- **Time Horizon:** Each experiment was targeted for 15 rounds of interaction.
- **Domain Diversity:** Ten meta-prompt scenarios spanning creative writing, data science, web scraping, file organization, image processing, personal finance, research assistance, text analysis, code generation, and simulation/modeling.
- **Initial State:** All agents began with identical primitive tools and no pre-defined specializations.
- **Incentive Structure:** The Tool Complexity Index (TCI) was heavily weighted toward composition:
$$\text{TCI} = 0.5 \cdot C_{\text{code}} + 1.0 \cdot C_{\text{iface}} + 10.0 \cdot C_{\text{comp}}$$
This 20:1 ratio between compositional and code complexity created strong selective pressure for tool collaboration.

A.2 Cross-Scenario Analysis

Analysis of the final tool ecosystems reveals three distinct patterns of emergence across domains.

A.2.1 Universal Emergence of Domain-Relevant Toolchains

Across all ten scenarios, agents demonstrated remarkable domain awareness, immediately creating tools highly relevant to their assigned meta-prompt. This suggests that the large language model's pre-training provides sufficient domain knowledge to guide initial tool creation, even without explicit domain expertise.

A.2.2 Heterogeneous Collaboration Rates

The rate of tool composition (tools with $C_{\text{comp}} > 0$) varied dramatically across domains:

- **High Collaboration Domains:** Creative Writing (25.0%), Text Analysis (24.0%), Simulation/Modeling (23.8%)
- **Medium Collaboration Domains:** Image Processing (17.9%), Research Assistant (18.5%), Personal Finance (15.4%)
- **Low Collaboration Domains:** Data Science (11.1%), Code Generation (9.7%), File System (12.1%), Web Scraping (17.2%)

This variation suggests that certain problem domains naturally lend themselves to compositional approaches, while others favor monolithic tool architectures.

A.2.3 Systematic Environmental Adaptation

Remarkably, agents across multiple scenarios independently created "RateLimitMonitor" tools in response to API constraints. This meta-tool appeared in 7 out of 10 scenarios, demonstrating consistent environmental problem-solving capabilities that transcend domain boundaries.

A.3 Quantitative Results

Table 3 presents a comprehensive comparison of emergence patterns across all ten scenarios.

A.4 Key Findings

This comprehensive baseline study provides four critical insights into emergent intelligence in decentralized agent societies:

Table 3: Cross-Scenario Baseline Emergence Results

Scenario	Total Tools	Valid Tools	Success Rate (%)	Avg TCI	Max TCI	Compositional Tools (%)
Code Generation Toolkit	31	31	100.0	4.20	14.64	3 (9.7)
Creative Writing Assistant	27	24	88.9	4.69	10.69	6 (25.0)
Data Science Suite	19	18	94.7	4.64	11.33	2 (11.1)
File System Organizer	37	33	89.2	6.97	21.42	4 (12.1)
Image Processing Kit	31	28	90.3	5.73	14.09	5 (17.9)
Personal Finance Manager	27	26	96.3	5.23	21.66	4 (15.4)
Research Assistant Bot	31	27	87.1	4.59	12.80	5 (18.5)
Simulation and Modeling	24	21	87.5	4.99	11.26	5 (23.8)
Text Analysis Tools	27	25	92.6	5.00	17.05	6 (24.0)
Web Scraping Utilities	34	29	85.3	7.60	19.02	5 (17.2)
Overall	**288**	**262**	**91.0**	**5.45**	**21.66**	**45 (17.2)**

1. **Domain-Agnostic Emergence:** All scenarios demonstrated immediate, relevant tool creation, suggesting that emergent specialization is robust across problem domains. The 91.0% overall success rate indicates that the core emergence mechanisms are reliable and generalizable.

2. **Composition as an Advanced Skill:** Tool composition occurred in only 17.2% of successful tools across all scenarios. However, compositional tools consistently achieved higher complexity scores, with the highest TCI scores in each domain typically belonging to compositional tools. This confirms that while composition is rare, it produces substantially more valuable outputs.

3. **Domain-Dependent Collaboration Propensity:** The 2.6x variation in collaboration rates between domains (9.7% to 25.0%) suggests that certain problem structures naturally encourage compositional thinking. Creative and analytical domains showed higher collaboration rates than technical implementation domains.

4. **Consistent Environmental Problem-Solving:** The spontaneous emergence of "RateLimitMonitor" tools across 7 scenarios demonstrates that agent societies can identify and address systemic constraints that are orthogonal to their primary objectives. This meta-cognitive capability is a strong indicator of robust collective intelligence.

A.5 Implications for Future Research

These baseline results establish clear benchmarks for measuring the impact of advanced interaction protocols. Future experiments incorporating Boids-inspired dynamics, explicit communication mechanisms, or evolutionary selection should be evaluated against these baseline collaboration rates and complexity distributions.

The observed domain-dependent variation in collaboration propensity also suggests that different meta-prompt scenarios may serve as more sensitive indicators of emergent collaboration. Creative Writing and Text Analysis scenarios, with their high baseline collaboration rates, may be particularly valuable for detecting subtle improvements in compositional behavior.

B Baseline Emergence: Data Science Suite (10 Agents, 5 Rounds)

This section reports results for experiment `exp1_baseline_emergence_data_science_suite_20250902_123439`. We analyze system-level productivity, test outcomes, and Tool Complexity Index (TCI) dynamics over rounds, and distill fine-grained insights from agent reflection histories.

Metric	Value
Agents	10
Rounds	5
Total tools created	50
Final tools in system	24
Tools per round	10.0
Total tests created	50
Tests passed	41
Tests failed	9
Test pass rate	82.00%
Testing coverage	100.00%
Collaboration events	0
Events per round	0.0

Table 4: Experiment summary for `exp1_baseline_emergence_data_science_suite_20250902_123439`.

Round	Avg TCI	Avg Code	Avg Interface	Avg Compositional	Tools
1	5.4081	9.0042	0.9060	0.0000	6
2	6.2170	9.4721	0.8977	0.0583	12
3	6.4830	8.5141	0.9135	0.1313	16
4	6.4295	8.1790	0.9400	0.1400	20
5	5.9879	7.7396	0.9514	0.1167	24

Table 5: Complexity evolution across rounds (Tool Complexity Index and its components).

B.1 Summary Metrics

B.2 Complexity Dynamics Over Rounds

Key trends. (1) **TCI rises then eases:** Avg TCI increases from 5.41 (R1) to 6.48 (R3), then softens to 6.43 (R4) and dips to 5.99 (R5). (2) **Code complexity declines:** Avg code complexity steadily decreases after R2 (9.47 $\rightarrow$ 7.74), consistent with refactoring/simplification as the toolset matures. (3) **Interface robustness inches up:** Avg interface complexity rises (0.906 $\rightarrow$ 0.951), indicating more consistent interfaces and/or improved probe success. (4) **Moderate composition:** Compositional complexity grows to R4 (0.14) and slightly eases (0.117), suggesting increasing but not pervasive composition. (5) **Strong testing discipline:** 100% coverage with 82% pass rate (41/50); mid/late-round failures concentrate as scope broadens.

B.3 Fine-Grained Insights from Agent Reflections

Agents repeatedly identify an ecosystem gap: deep, end-to-end pipelines that chain cleaning, transformation, analysis, and visualization. Sample reflections:

- **Agent_01**: *"Automated, end-to-end data science workflows... a comprehensive, modular Deep Data Science Workflow tool is absent."* Proposed: an *End-to-End Data Science Pipeline* integrating cleaning, feature transforms, exploratory analysis, and viz prep.
- **Agent_02**: *"Combine DataCleaner, DataPreprocessingPipeline, and DataAnalysisPipeline into a higher-level, 'deep' composite workflow."* Emphasizes chaining reliable modules for reusability and scale.
- **Agent_04**: *"Missing integrated tools for advanced data validation, anomaly detection, and systematic error handling."* Proposed: an *Automated Data Validation and Anomaly Detection Pipeline*.
- **Agent_06**: *"Unified Data Preparation and Modeling Workflow... cleans, engineers features, fits simple models, and outputs diagnostics."* Reflects gradual shift from basic preprocessing to modeling orchestration.

Interpretation. The rise in interface complexity and modest compositional gains, alongside declining code complexity, matches the reflection-driven shift from single-purpose utilities to orchestrated pipelines. As teams standardize interfaces and compose stable building blocks, average code

complexity per tool falls (less bespoke logic), while system capability deepens through composition—consistent with the observed R1–R3 TCI rise and later plateau as complexity diversifies across many smaller, interoperable tools.

Reproducibility notes. Results are computed from `results.json` (per-round complexity in `complexity_over_rounds`) and summarized in `summary.txt`. Experiment directory: `experiments/exp1_baseline_emergence_data_science_suite_20250902_123439`.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

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 abstract and introduction claim a framework that unifies Boids style local rules with evolutionary adaptation and reports emergent coordination and specialization. The methods and results implement this framework and support these claims within the stated 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: We clearly note short horizons from API limits, small populations, Python-centric domains, and single-seed analyses. We also state that TCI needs external validation, which bounds generalization.

   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. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - 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. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. 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: We provide numbered equations that fully define all constructs and assumptions, plus a brief justification of the convex combinations and max-scaling. There are no theorems, but definitions are complete and checkable.

   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.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

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: We describe the agent loop, observables, default hyperparameters, evaluation protocol, and implementation notes. We point to per-run logs, JSON traces, and directories that reproduce tables and 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, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) 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: We include structured logs, JSON traces, and per-run directories referenced in the paper with instructions to rebuild figures and tables from these artifacts. These anonymized materials accompany the submission to enable faithful reproduction.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) 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. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

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: We specify agent counts, rounds, topologies, and default thresholds (k=5, $_{s}ep = 0.45$,$_{a}li = 2/3$,$_{c}oh = 0.6$).$We also define the metrics and procedures used across all experiments. Guidelines:$

   7. The answer NA means that the paper does not include experiments.
   8. 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.
   9. The full details can be provided either with the code, in appendix, or as supplemental material.

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: We run replicated experiments across randomized initializations and network topologies and report per-run metrics. The released logs allow computing confidence intervals or error bars if needed.

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, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

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: [Yes]

Justification: We report population sizes and horizons, note LLM API usage and rate-limit effects, and provide run-time artifacts. These details let readers estimate compute and time requirements.

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.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: No human subjects or sensitive data are used. All tools run in a sandbox with safeguards such as recursion limits and automated testing.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

Broader impacts

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

Answer: [Yes]

Justification: We discuss positive uses for alignment, governance, and evolving tool ecosystems. We also note risks such as collapse dynamics and describe mitigations via sandboxing and quality control.

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 (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: We describe execution safeguards including sandboxing, recursion limits, automated tests, error logging, and controlled visibility. These measures reduce misuse risk for released artifacts.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite prior works and list external dependencies at the tool level. Any third-party assets are used under their original licenses and will be credited accordingly.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We introduce a prompt-executable Boids constraint suite and a TCI analyzer with clear equations, I/O definitions, and implementation notes. Structured logs and run folders document these assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Not applicable: the work uses autonomous software agents and synthetic artifacts only. No crowdsourcing or human studies were conducted.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Not applicable: there were no human subjects or user studies, so no IRB review was required.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We describe how LLM agents follow an observe, reflect, build loop and are constrained by prompt-executable validators. This usage is central to the method and is documented in Methods with related citations.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.