Benchmarking LLMs' Swarm intelligence

Kai Ruan[†], **Mowen Huang**[†], **Ji-Rong Wen, Hao Sun**[*]
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China
{kairuan, retr0, jrwen, haosun}@ruc.edu.cn

Abstract

Large Language Models (LLMs) show potential for complex reasoning, yet their capacity for emergent coordination in Multi-Agent Systems (MAS) when operating under strict constraints—such as limited local perception and communication, characteristic of natural swarms—remains largely unexplored, particularly concerning the nuances of swarm intelligence. Existing benchmarks often do not fully capture the unique challenges of decentralized coordination that arise when agents operate with incomplete spatio-temporal information. To bridge this gap, we introduce **SwarmBench**, a novel benchmark designed to systematically evaluate the swarm intelligence capabilities of LLMs acting as decentralized agents. SwarmBench features five foundational MAS coordination tasks (Pursuit, Synchronization, Foraging, Flocking, Transport) within a configurable 2D grid environment, forcing agents to rely primarily on local sensory input ($k \times k$ view) and local communication. We propose metrics for coordination effectiveness and analyze emergent group dynamics. Evaluating several leading LLMs (e.g., `deepseek-v3`, `o4-mini`) in a zero-shot setting, we find significant performance variations across tasks, highlighting the difficulties posed by local information constraints. While some coordination emerges, results indicate limitations in robust planning and strategy formation under uncertainty in these decentralized scenarios. Assessing LLMs under swarm-like conditions is crucial for realizing their potential in future decentralized systems. We release SwarmBench as an open, extensible toolkit—built upon a customizable and scalable physical system with defined mechanical properties. It provides environments, prompts, evaluation scripts, and the comprehensive experimental datasets generated, aiming to foster reproducible research into LLM-based MAS coordination and the theoretical underpinnings of Embodied MAS. Our code repository is available at `https://github.com/x66ccff/swarmbench`.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in language understanding and generation [1], leading to growing interest in deploying them as autonomous agents capable of perception, tool use, and collaboration [2, 3]. Consequently, research is increasingly investigating the collaborative potential of LLM-driven agents, particularly in tasks requiring spatial reasoning and interaction, connecting to broader studies of collective intelligence in artificial and human systems [4]. However, current evaluations predominantly focus on individual agent skills or multi-agent scenarios characterized by ample communication, global visibility, or predefined organizational structures [5–7]. Such settings often sidestep the fundamental challenge of achieving coordination when agents operate under severe, decentralized constraints.

[†]Equal contribution.
[*]Corresponding author.

Preprint. Under review.

Inspired by decades of research across biology, physics, and robotics, a critical question remains largely unexplored in the LLM context: Can effective coordination and collective intelligence emerge from the decentralized actions of numerous LLM agents operating with strictly limited perception and communication, akin to natural swarms? This principle forms the bedrock of Swarm Intelligence, which investigates how complex group behaviors arise from simple, local interactions [8]. Nature provides compelling examples, from army ants forming living structures [9, 10] and locusts achieving coordinated marching [11], to microswimmers forming vortices [12]. Seminal simulations, like Reynolds' flocking model [13], demonstrated that sophisticated global patterns can emerge purely from local rules. This paradigm, emphasizing decentralization, local sensing, and minimal communication, has been successfully applied in Swarm Robotics, where collectives of simple robots achieve complex tasks like shape formation [14, 15]. Therefore, a key unknown is whether LLMs, despite their advanced cognitive capabilities, can effectively participate in such decentralized swarms, bridging individual sophistication and emergent collective action under classic swarm constraints.

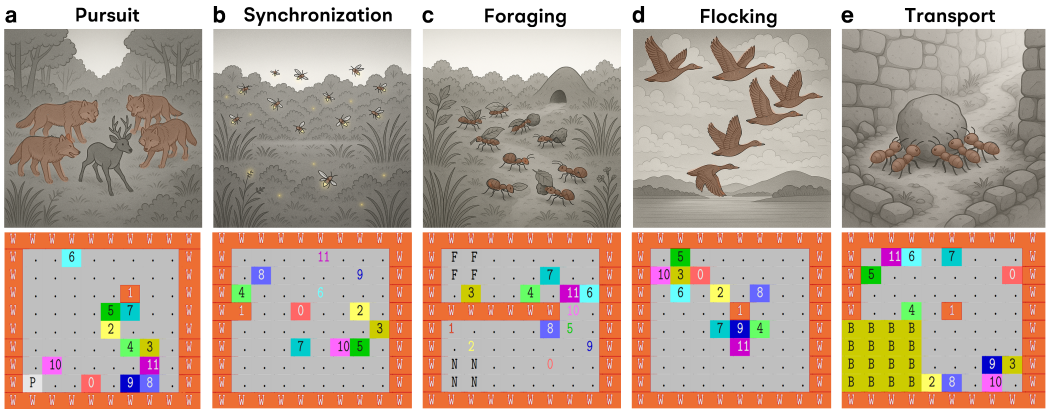


Figure 1: **Swarm Intelligence: Natural Inspiration and SwarmBench Tasks. Top row:** Examples of collective behavior in nature driven by local interactions: (a) cooperative wolf pursuit, (b) firefly synchronization, (c) ant foraging [9, 10], (d) bird flocking [13], and (e) cooperative ant transport. **Bottom row:** Corresponding abstract tasks simulated in SwarmBench's 2D grid environment, depicting agents (represented by colored squares) facing analogous coordination challenges involving targets (`P`), food (`F`), nests (`N`), and obstacles (`B`), constrained by walls (`W`). Agents rely solely on *local* perception and communication, providing a testbed for emergent decentralized coordination.

Existing benchmarks do not adequately address this question rooted in the classical swarm intelligence paradigm. Some evaluate spatial reasoning fundamentals without multi-agent dynamics [16], while others test single-agent reasoning in complex tasks [17]. Multi-agent benchmarks often employ structured games [18], collaborative tasks with rich communication [5, 6, 19], or scenarios where coordination structures are imposed rather than emerging organically [20]. While valuable, these approaches often do not center on the core challenge of achieving robust *decentralized coordination* despite *severe limitations on perception and communication*. Consequently, whether LLM collectives can exhibit complex swarm phenomena—like spontaneous leadership [20], the role of noise/diversity [21, 22], or information cascades—under such stringent, decentralized conditions is a critical open question. Existing approaches often bypass the core constraints necessary to observe such organically emerging complexities, highlighting a gap SwarmBench aims to address.

To fill this gap, we introduce SwarmBench, a benchmark specifically designed to evaluate the emergent coordination capabilities of LLM agents acting as individuals in a decentralized swarm. Inspired by benchmarks like SnakeBench [23] and ARC-AGI [24], SwarmBench presents five fundamental multi-agent coordination challenges—Pursuit, Synchronization, Foraging, Flocking, and Transport—within a flexible 2D grid world. Critically, agents operate with restricted perception (a small local view) and minimal, optional local communication, forcing reliance on local cues and implicit coordination. We propose metrics to quantify task success and efficiency, and the nature of the emergent collective behavior, including measures related to behavioral diversity.

The SwarmBench framework (Figure 2) was used to conduct extensive zero-shot evaluations of several prominent LLMs. Our contributions are:

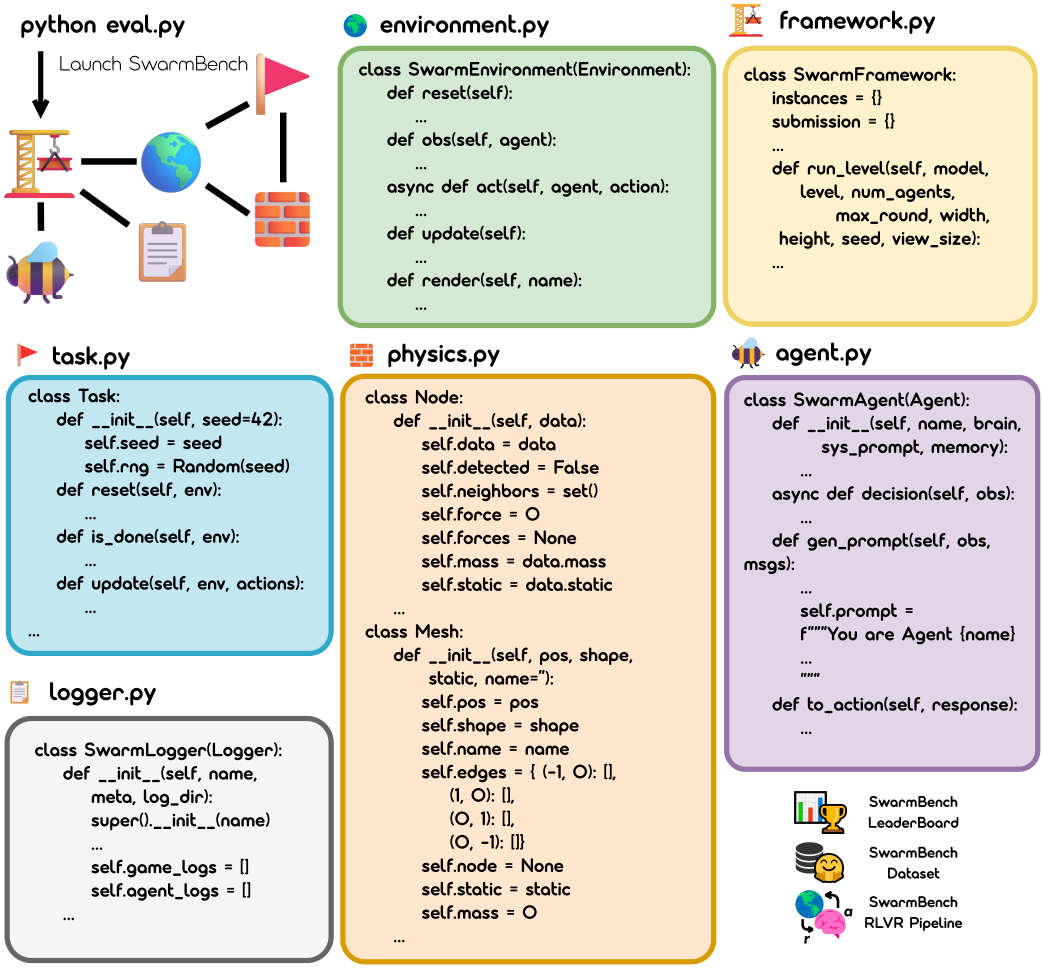


Figure 2: **Conceptual Architecture of SwarmBench.** The SwarmBench framework is designed for modularity and extensibility in evaluating LLM-based swarm intelligence. An evaluation begins by launching SwarmBench, which orchestrates the core interactions between the defined task, the simulation environment, the LLM-powered agents, and a comprehensive logger for data capture. Snippets from underlying modules illustrate how SwarmBench implements this: defining diverse coordination tasks, managing the environment and its physics, structuring agent perception and decision-making logic, providing the overall experimental framework, and logging results. This architecture enables systematic benchmarking, generates comprehensive Datasets from experiments, and facilitates the creation of a LeaderBoard for model comparison. The framework is also designed to readily support future extensions, such as the integration of an RLVR Pipeline.

- **SwarmBench**: A novel benchmark grounded in swarm intelligence principles, designed to assess emergent decentralized coordination in LLM swarms under strict perception and communication constraints.
- A systematic **evaluation** of contemporary LLMs on SwarmBench, characterizing their current abilities and limitations in canonical swarm scenarios.
- An **analysis** of emergent group dynamics, connecting LLM swarm behavior (e.g., behavioral variability, failure modes) to established collective intelligence concepts.
- An **open-source toolkit**, built upon a customizable and scalable physical system with defined mechanical properties, comprising environments, standardized prompts, evaluation scripts, and the comprehensive datasets generated, to facilitate reproducible research into LLM-based swarm intelligence.

Our findings indicate that while LLMs exhibit potential for basic coordination, they struggle significantly with long-range planning and robust spatial reasoning under uncertainty when operating under severe decentralization. SwarmBench provides a dedicated platform to measure progress and guide future research towards developing LLMs capable of genuine collective intelligence in decentralized settings. Understanding such capabilities is vital, given the growing focus on collective behavior in artificial and human systems [25, 26].

2 Related Work

Our research builds upon the foundations of swarm intelligence, multi-agent systems (MAS), and LLMs. We investigate classical swarm principles using modern LLMs, positioning our work relative to recent multi-agent evaluation methodologies.

Swarm Intelligence and Self-Organization Swarm intelligence examines how complex, adaptive group behaviors emerge from local interactions among individuals with relatively simple capabilities, drawing from natural systems like insect colonies, bird flocks, and microscopic swimmers [8, 13, 12]. Biological examples include army ants constructing living bridges [9, 10] and locusts transitioning to coherent marching [11]. This natural blueprint inspired swarm robotics, focusing on coordinating large groups of robots, often with limited individual abilities [27]. Seminal projects like Kilobots demonstrated collective shape formation by a thousand simple robots using only local communication [14, 15]. Further advancements include adaptive control hierarchies like Self-Organizing Neural Systems (SoNS) [28], and specialized simulators like ARGoS [29] and Kilombo [30]. The functional role of diversity or noise, potentially enhancing coherence [22, 21], resonates with our exploration. SwarmBench adopts core operational constraints (local sensing, minimal communication) but substitutes simpler agents with powerful LLMs, to investigate how swarm intelligence manifests with sophisticated cognitive entities. This focus on physical action differentiates our work from data processing approaches like Swarm Learning [31].

LLM as Agents in Multi-Agent Systems Utilizing LLMs as the decision-making core for autonomous agents is a rapidly growing field [2, 32]. LLMs bring vast world knowledge and reasoning, enabling more adaptable agent interactions [33]. This extends to MAS, where LLMs might enhance communication and teamwork [34, 35], reflecting human collective intelligence factors [4]. LLMs are applied in diverse multi-agent contexts: software development (`MetaGPT` [35]), simulated scientific discovery [36, 37], complex social simulations [33, 3, 38, 39], and code generation [40]. Studies on emergent cooperation and Theory of Mind in LLM teams identify promising capabilities alongside consistency limitations [41]. Imposed organizational structures can improve efficiency [20]. LLM integration into multi-robot systems is also surveyed [42]. However, many investigations involve rich communication channels, predefined roles, or assume reliable information transfer, contrasting with swarm systems where noise and limited propagation are defining features [43]. A key unaddressed question is how LLMs perform in large, decentralized groups where coordination emerges from local perception and constrained signaling. SwarmBench directly targets this.

Benchmarking LLM Coordination and Spatial Reasoning Meaningful evaluation of LLM agents in MAS requires appropriate benchmarks. Recent efforts use cooperative games: LLM-Coordination (Hanabi, Overcooked [18]), Collab-Overcooked (natural language in Overcooked [6]),

and COBLOCK (3D construction [44]). SnakeBench uses competitive games [23] without strict swarm constraints. These game-based benchmarks often provide full visibility or structured communication distinct from swarm intelligence's local-information world. Other benchmarks explore complex tasks: MultiAgentBench (MARBLE) with diverse scenarios, often with roles [5]; VillagerBench (Minecraft group tasks [19]); and Generative Agents (emergent social dynamics [33]). These showcase LLM capabilities but typically rely on higher-level coordination mechanisms. Foundational reasoning is assessed by SpatialEval (spatial understanding [16]) and BALROG (single-player agentic reasoning [17]). LLMs reportedly struggle with patterns like multi-agent flocking [45]. SwarmBench distinguishes itself by concentrating on emergent decentralized coordination within LLM swarms, adopting classical swarm intelligence constraints (restricted perception/communication). It employs foundational, nature-inspired tasks and analyzes collective dynamics often overlooked in benchmarks focused on structured frameworks.

LLM-Driven Coordination in Embodied Simulations Significant research explores LLMs for coordination within embodied multi-agent systems, often using sophisticated 3D simulators (e.g., AI-THOR, Habitat) for tasks like household assistance and navigation [46–48]. These systems frequently address challenges like heterogeneous capabilities, task allocation [49], and integrating perception with planning [50, 51], alongside developing modular architectures [52], world models [53], or deadlock resolution [54]. While demonstrating progress, this body of work typically operates with richer sensory inputs, more sophisticated communication (sometimes low-distortion [43]), or different architectural assumptions (e.g., centralized components). In contrast, SwarmBench specifically isolates and evaluates the emergence of fundamental swarm intelligence driven by *minimal local interactions* under highly constrained, decentralized conditions, reflecting classical swarm intelligence principles. By focusing on these aspects in a simplified 2D grid world, SwarmBench provides a complementary evaluation focused on raw emergent coordination potential from local constraints.

3 SwarmBench

To evaluate the capacity of Large Language Models (LLMs) for emergent decentralized coordination under constraints typical of swarm intelligence, we introduce SwarmBench. This benchmark provides a suite of multi-agent tasks within a configurable 2D grid-world environment, coupled with standardized protocols for evaluating LLM-driven agents. SwarmBench focuses on scenarios where agents possess only limited local perception and rely on local communication capabilities, necessitating the emergence of collective strategies from decentralized interactions rather than global planning or explicit centralized control. Further details on the environment, agent capabilities, and evaluation protocol are provided in Appendix B.

3.1 Environments

SwarmBench utilizes a simulation environment based on a 2D grid world, a design choice aligned with foundational AI benchmarking (e.g., SnakeBench [23], ARC-AGI [24]) to facilitate focused investigation of core coordination dynamics. This environment is a customizable physical system with explicitly modeled mechanical interactions (detailed in Appendix A). The benchmark includes five core multi-agent coordination tasks: **Pursuit**, **Synchronization**, **Foraging**, **Flocking**, and **Transport**. These tasks, visualized in Figure 1 and detailed in Appendix B, probe different facets of emergent swarm behavior. The environment framework is extensible and supports procedural generation of instances to ensure robust evaluation.

3.2 Observations, Actions, and Communication

Agents operate with restricted perception, primarily an egocentric $k \times k$ grid view, and can engage in optional local, anonymous message passing. Based on their local observation (which includes the grid view, self-status, and received messages), agents decide on a primary action (e.g., movement, task-specific) and an optional message. This setup compels reliance on local cues and implicit coordination. Specifics of the observation packet, action space, physics of movement, and communication mechanics are elaborated in Appendix B, with the agent prompt structure in Appendix C.

3.3 Evaluation Setting and Models

We employ a zero-shot evaluation protocol where each agent is controlled by an independent, stateless LLM instance. Persistence of memory is managed through the prompt. SwarmBench is model-agnostic; our experiments (Section 4) utilize several contemporary LLMs without task-specific fine-tuning. The detailed evaluation methodology is described in Appendix B.

3.4 Evaluation Metrics for Group Dynamics

To quantitatively analyze emergent collective behaviors, we compute metrics based on agent positions and actions, capturing aspects like behavioral diversity and movement coordination. These metrics, detailed in Appendix E, facilitate the analysis of emergent strategies and performance.

4 Results

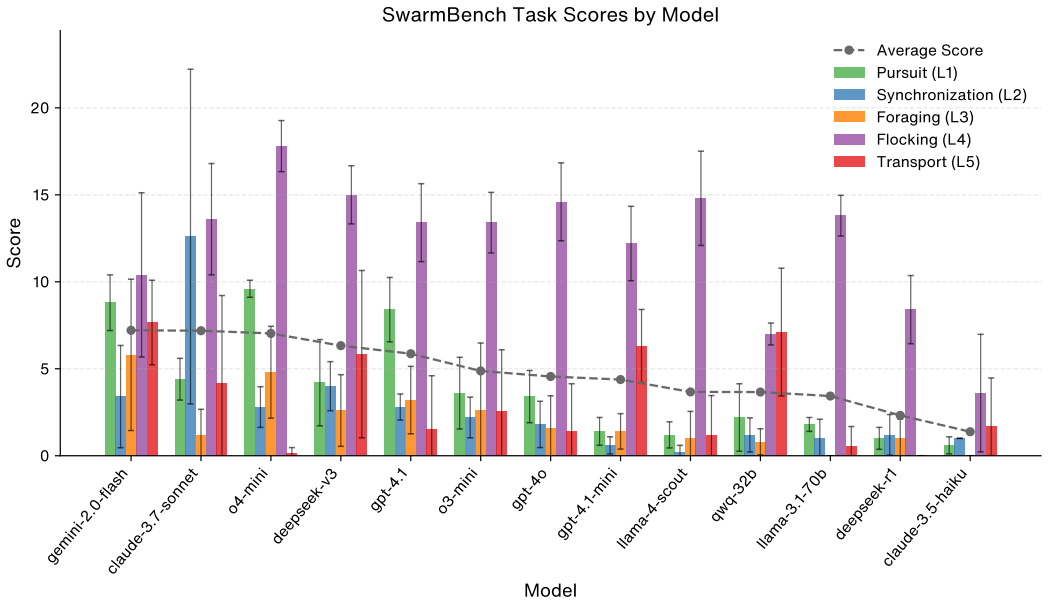


Figure 3: **Overview of LLM Performance on SwarmBench Tasks.** Average scores achieved by different LLMs across the five core tasks (`Pursuit`, `Synchronization`, `Foraging`, `Flocking`, `Transport`). Bars represent the mean score over 5 simulation runs. Performance varies significantly depending on the model and the specific coordination challenge. Detailed scores and standard deviations are provided in Table S.1 in Appendix F

.

We evaluated thirteen contemporary LLMs (Fig. 3) on the five core SwarmBench tasks (`Pursuit`, `Synchronization`, `Foraging`, `Flocking`, `Transport`) under the zero-shot protocol. Agents operated with a 5×5 local view ($k = 5$), making decisions based on this restricted perception and the potential for local communication via the `MSG` action (as detailed in Section B.2). Performance, averaged over 5 simulation runs per model per task, reveals significant variation based on both the model and the specific coordination challenge, highlighting the difficulty of decentralized coordination under strict local constraints.

4.1 Task Performance Comparison

Figure 3 visually summarizes the average task scores achieved by the evaluated LLMs across the five core SwarmBench challenges: `Pursuit`, `Synchronization`, `Foraging`, `Flocking`, and `Transport`. The results reveal significant performance variability, contingent on both the specific LLM and the nature of the coordination task.

Overall trends indicate that tasks presented varying levels of difficulty, with `Flocking` generally yielding the highest scores across models, while `Synchronization` showed greater divergence in performance. Model strengths also differed considerably; for instance, `gemini-2.0-flash` and `o4-mini` demonstrated relative strength in spatial tasks like `Pursuit` and `Foraging`, whereas `claude-3.7-sonnet` excelled specifically in `Synchronization`. Notably, no single model dominated all tasks, and several models (`deepseek-r1`, `claude-3.5-haiku`) struggled significantly across the board in this zero-shot setting, underscoring the inherent difficulty of these swarm coordination problems. Figure 4 further illustrates the score progression dynamics over time.

These results emphasize that LLM coordination ability under swarm-like constraints is highly task-dependent and relies heavily on emergent strategies formed from local information. The observed variability points to diverse capabilities and limitations among current models when faced with decentralized coordination challenges. For detailed numerical results, including mean scores and standard deviations over the 5 simulation runs per model and task, please refer to Table S.1 in Appendix F.

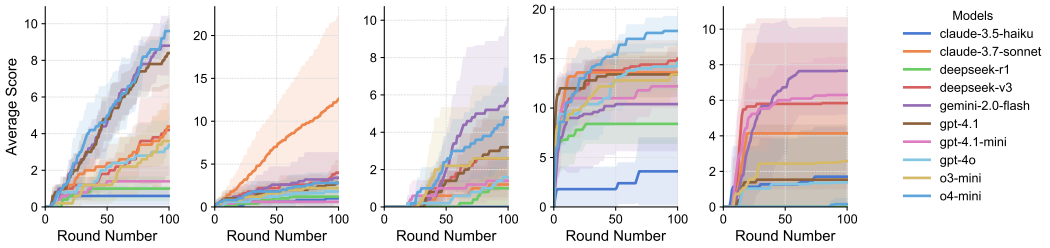


Figure 4: **LLM Score Progression on SwarmBench Tasks Over Time.** Average task score accumulation over 100 simulation rounds for different LLMs across the five core tasks (`Pursuit`, `Synchronization`, `Foraging`, `Flocking`, `Transport`). Lines represent the mean score trajectory, and shaded areas indicate the standard deviation across 5 simulation runs. This illustrates the dynamics of performance development during each task.

4.2 Analysis of Emergent Group Dynamics and Communication Correlates

To understand the behavioral underpinnings of performance across all tasks, we conducted two main analyses on the simulation data.

First, we performed a primary analysis focusing on the correlation between physical group dynamics metrics (defined in Section 3.4 and detailed in Appendix E) and the final task scores, aggregating data from all 325 simulation runs. This analysis aimed to identify general principles of effective swarm coordination emerging from agents' physical actions and local observations. The detailed results of this dynamics analysis are presented in Appendix H. Key findings (Table S.2) indicate that higher scores are significantly correlated with behavioral *variability* (e.g., `std_action_entropy`, $r = 0.300$; `std_dominant_action_prop`, $r = 0.274$) and *efficiency* (e.g., `prop_stay_actions`, $r = 0.297$). Conversely, excessive movement (`prop_move_actions`, $r = -0.222$) and persistent alignment (`avg_polarization_index`, $r = -0.241$) are negatively correlated with performance. Visualizations are provided (Figures S.8, S.9 in Appendix H), and a linear regression model based solely on these dynamics features explains approximately 24.5% of score variance (Table S.3 in Appendix H), emphasizing the primary role of emergent physical coordination.

Second, to investigate the potential contribution of explicit communication, we conducted a supplementary analysis focused on messages generated by agents via the `MSG` action. We quantified basic communication patterns (frequency, length) and assessed semantic properties using standard NLP techniques (methodology detailed in Appendix G). Correlating these communication-related metrics with task scores revealed weaker relationships than observed for physical dynamics (detailed results in Appendix I). The statistically significant findings from this sampled analysis were:

- **Message Length:** A weak positive correlation between the average length of non-empty messages per run (`avg_non_empty_msg_length_run`) and score ($r \approx 0.19$, $p < 0.001$).

- **Semantic Stability:** A weak negative correlation between the variability (standard deviation) of semantic similarity among messages within a run (`std_similarity_run`) and score ($r \approx -0.17$, $p < 0.01$).

Message frequency and average semantic similarity did not show significant correlations in this supplementary analysis (see Appendix I). These findings suggest that while physical dynamics predominantly drive performance, characteristics of the communication itself—specifically, message length and semantic consistency—may exert a minor influence.

4.3 Visualization of Emergent Behaviors

Visual examples of agent trajectories and interactions for each core task are provided in Appendix D (Figures S.2 through S.6). These visualizations illustrate the challenges faced by agents, such as forming effective containment in `Pursuit`, achieving synchrony in `Synchronization`, navigating efficiently in `Foraging`, maintaining cohesion in `Flocking`, and coordinating pushes in `Transport`. They also qualitatively reveal differences in strategies and success levels across different models and runs, complementing the quantitative analysis.

4.4 Analysis of Failure Modes

Qualitative observation of simulation runs, particularly those resulting in low scores (see Appendix F for score distributions and Appendix D for visual examples), reveals common failure patterns. In tasks like `Pursuit` and `Transport`, agents often struggle with sustained coordination; initial promising formations may dissolve due to individual agents making suboptimal local decisions or failing to interpret implicit cues from neighbours. Cascading failures, where one agent's poor move disrupts others, were observed. In `Foraging`, agents sometimes exhibited inefficient exploration or got stuck in loops. The difficulty in robust planning under uncertainty, stemming from the limited local view, appears to be a major factor. These failures are also reflected in the dynamics analysis (Appendix H), where high negative correlations with score were found for metrics indicating disorganised or excessive movement (e.g., `prop_move_actions`).

5 Discussion

Our SwarmBench evaluations offer key insights into LLM-driven decentralized coordination. The primary analysis (Appendix H) reveals that emergent physical group dynamics—particularly behavioral *flexibility* (e.g., `std_action_entropy`) and *efficiency* (e.g., `prop_stay_actions`)—are significant drivers of performance, collectively explaining approximately 24.5% of score variance. In contrast, while our supplementary communication analysis (Appendices G, I) suggests that characteristics like message length and semantic consistency have a statistically significant, albeit weaker, positive correlation with outcomes, a substantial portion of performance variation remains. This indicates that factors intrinsic to the LLMs themselves—such as differences in their pre-trained reasoning capabilities, their proficiency in spatial understanding derived from textual inputs, or their interpretation of the task prompts (Appendix C)—are highly influential yet not fully captured by our current set of group dynamics metrics.

The evaluated LLMs, operating zero-shot under these strict local constraints, appear to compensate for what might be underdeveloped or inefficient explicit communication strategies by relying more heavily on adaptive physical behaviors and implicit coordination cues. This approach differs notably from many natural swarm systems, which often achieve robust coordination through efficient, low-distortion signaling pathways [43]. The difficulties LLM swarms face in maintaining sustained collective alignment and robust planning under uncertainty likely stem from challenges in spatial reasoning and effective conflict resolution when information is severely limited.

Furthermore, the sensitivity analysis (Appendix J) highlights that performance is non-trivially dependent on environmental parameters like agent density (N) and perception range (k). This underscores the challenge of developing LLM agents that are not only capable of coordination but are also robustly adaptable to varying information availability and group dynamic conditions. Understanding and overcoming these limitations are crucial for harnessing the full potential of LLMs in decentralized, multi-agent systems.

6 Conclusion

In this work, we introduced SwarmBench, a novel benchmark specifically designed to assess the emergent decentralized coordination capabilities of Large Language Models operating under conditions characteristic of swarm intelligence. Our evaluations reveal that while contemporary LLMs demonstrate foundational abilities for basic coordination within SwarmBench tasks, they face significant challenges in achieving the sophisticated and robust collective behaviors observed in natural systems. Particularly, their emergent collective strategies, especially concerning information propagation and resilience to noise under strict decentralization, do not yet mirror the highly efficient, low-distortion signaling and coordinated action typical of, for instance, bird flocks [43].

This observed gap between individual LLM sophistication and emergent collective intelligence under decentralized constraints highlights a critical direction for future research. Developing LLM-based systems that can more effectively bridge this divide is paramount. SwarmBench offers a systematic platform to measure progress in this domain, guiding the development of LLMs towards more robust, adaptive, and genuinely collective behavior. Such capabilities will be increasingly vital as AI systems become more deeply integrated into complex, real-world contexts requiring decentralized coordination, from autonomous robotic swarms to distributed computational networks.

7 Limitations

It is important to acknowledge the scope defined by SwarmBench's current design. The adoption of a 2D grid world—though the SwarmBench engine is a customizable physical system with defined mechanics (Appendix A)—is a deliberate choice aligning with foundational AI benchmarks [23, 24]. This facilitates focused investigation of core coordination dynamics but necessarily abstracts from the complexities of continuous 3D spaces relevant to physical swarm robotics. Our primary focus on zero-shot evaluation provides a crucial baseline for LLMs' intrinsic abilities but defers exploration of adaptive learning mechanisms. The sensitivity analysis (Appendix J) indicates that results can vary with parameters like agent count (N) and field of view (k), implying that findings are contingent on specific configurations. Furthermore, our communication analysis employed a particular Sentence-BERT model; a broader suite of NLP techniques might yield additional insights. These design choices establish clear boundaries, enabling SwarmBench to serve as a foundational platform for systematically dissecting emergent coordination under the specific, well-defined constraints central to classical swarm intelligence.

8 Future Work

Building on SwarmBench's framework and our initial findings, future work will focus on several key areas. These include exploring agent adaptation through learning mechanisms like reinforcement learning or fine-tuning, potentially integrating an RLVR Pipeline (e.g., [55–57]), and extending the benchmark with 3D environments, more complex physics, and novel tasks. Deeper investigations into inter-agent communication—analyzing diverse protocols, studying learned messaging, and employing broader NLP methods—are planned. Additionally, developing novel agent architectures and prompting strategies (cf. Appendix C) to enhance decentralized reasoning under constraint, alongside theoretical models for LLM-based swarms, will be crucial for advancing the field.

9 Broader Impacts

As research into decentralized AI systems like LLM-driven swarms progresses, spurred by tools such as SwarmBench, it is imperative to proactively consider the broader societal implications. The enhanced capabilities under development could, if deployed without robust ethical guidelines and safeguards, be misused for purposes such as automated surveillance, sophisticated disinformation campaigns, or the disruption of critical infrastructure. Consequently, technical advancements must be paralleled by dedicated research into mechanisms for controllability, transparency in agent and swarm behavior, and the steadfast alignment of these complex AI systems with human values and societal norms to ensure their responsible development and deployment.

References

[1] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.

[2] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101, 2025.

[3] Chen Gao, Xiaochong Lan, Nian Li, Yuan Yuan, Jingtao Ding, Zhilun Zhou, Fengli Xu, and Yong Li. Large language models empowered agent-based modeling and simulation: A survey and perspectives. *Humanities and Social Sciences Communications*, 11(1):1–24, 2024.

[4] Anita Williams Woolley, Christopher F. Chabris, Alex Pentland, Nada Hashmi, and Thomas W. Malone. Evidence for a Collective Intelligence Factor in the Performance of Human Groups. *Science*, 330(6004):686–688, October 2010. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1193147.

[5] Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong Wang, Cheng Qian, Xiangru Tang, Heng Ji, et al. MultiAgentBench: Evaluating the collaboration and competition of LLM agents. *arXiv preprint arXiv:2503.01935*, 2025.

[6] Haochen Sun, Shuwen Zhang, Lei Ren, Hao Xu, Hao Fu, Caixia Yuan, and Xiaojie Wang. Collab-Overcooked: Benchmarking and evaluating large language models as collaborative agents. *arXiv preprint arXiv:2502.20073*, 2025.

[7] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*, 2(4):6, 2023.

[8] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm intelligence: from natural to artificial systems*. Number 1. Oxford University Press, 1999.

[9] Chris R. Reid, Matthew J. Lutz, Scott Powell, Albert B. Kao, Iain D. Couzin, and Simon Garnier. Army ants dynamically adjust living bridges in response to a cost–benefit trade-off. *Proceedings of the National Academy of Sciences*, 112(49):15113–15118, 2015. doi: 10.1073/pnas.1512241112.

[10] Matthew J. Lutz, Chris R. Reid, Christopher J. Lustri, Albert B. Kao, Simon Garnier, and Iain D. Couzin. Individual error correction drives responsive self-assembly of army ant scaffolds. *Proceedings of the National Academy of Sciences*, 118(17):e2013741118, 2021. doi: 10.1073/pnas.2013741118.

[11] J. Buhl, D. J. T. Sumpter, I. D. Couzin, J. J. Hale, E. Despland, E. R. Miller, and S. J. Simpson. From disorder to order in marching locusts. *Science*, 312(5778):1402–1406, 2006. doi: 10.1126/science.1125142.

[12] Xiangzun Wang, Pin-Chuan Chen, Klaus Kroy, Viktor Holubec, and Frank Cichos. Spontaneous vortex formation by microswimmers with retarded attractions. *Nature Communications*, 14(1):186, 2023. doi: 10.1038/s41467-022-35427-7.

[13] Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, pages 25–34, 1987.

[14] Michael Rubenstein, Alejandro Cornejo, and Radhika Nagpal. Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198):795–799, 2014.

[15] Weixu Zhu, Sinan Oğuz, Mary Katherine Heinrich, Michael Allwright, Mostafa Wahby, Anders Lyhne Christensen, Emanuele Garone, and Marco Dorigo. Self-organizing nervous systems for robot swarms. *Science Robotics*, 9(96):eadl5161, 2024.

[16] Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, Yixuan Li, and Neel Joshi. Is A picture worth A thousand words? Delving Into Spatial Reasoning for Vision Language Models. In *The Thirty-Eighth Annual Conference on Neural Information Processing Systems*, 2024.

[17] Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wolczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, et al. BALROG: Benchmarking agentic LLM and VLM reasoning on games. *arXiv preprint arXiv:2411.13543*, 2024.

[18] Saaket Agashe, Yue Fan, Anthony Reyna, and Xin Eric Wang. LLM-coordination: evaluating and analyzing multi-agent coordination abilities in large language models. *arXiv preprint arXiv:2310.03903*, 2023.

[19] Yubo Dong, Xukun Zhu, Zhengzhe Pan, Linchao Zhu, and Yi Yang. Villageragent: A graph-based multi-agent framework for coordinating complex task dependencies in Minecraft. *arXiv preprint arXiv:2406.05720*, 2024.

[20] X Guo, K Huang, J Liu, W Fan, N Vélez, Q Wu, H Wang, TL Griffiths, and M Wang. Embodied LLM agents learn to cooperate in organized teams. arXiv 2024. *arXiv preprint arXiv:2403.12482*, 2024.

[21] Mohsen Raoufi, Pawel Romanczuk, and Heiko Hamann. Individuality in swarm robots with the case study of Kilobots: Noise, bug, or feature? In *ALIFE 2023: Ghost in the Machine: Proceedings of the 2023 Artificial Life Conference*. MIT Press, 2023.

[22] C. A. Yates, R. Erban, C. Escudero, I. D. Couzin, J. Buhl, I. G. Kevrekidis, C. C. Ioannou, P. Romanczuk, and D. J. T. Sumpter. Inherent noise can facilitate coherence in collective swarm motion. *Proceedings of the National Academy of Sciences*, 106(14):5464–5469, 2009. doi: 10.1073/pnas.0811195106.

[23] Greg Kamradt. Snake bench: Competitive snake game simulation with llms. `https://github.com/gkamradt/SnakeBench`, 2025.

[24] Francois Chollet, Mike Knoop, Gregory Kamradt, and Bryan Landers. ARC Prize 2024: Technical Report. *arXiv preprint arXiv:2412.04604*, 2025.

[25] Kang Hao Cheong and Michael C. Jones. Swarm intelligence begins now or never. *Proceedings of the National Academy of Sciences*, 118(42):e2113678118, 2021. doi: 10.1073/pnas.2113678118.

[26] Joseph B. Bak-Coleman, Mark Alfano, Wolfram Barfuss, Carl T. Bergstrom, Miguel A. Centeno, Iain D. Couzin, Jonathan F. Donges, Mirta Galesic, Andrew S. Gersick, Jennifer Jacquet, Albert B. Kao, Rachel E. Moran, Pawel Romanczuk, Daniel I. Rubenstein, Keren J. Tombak, Jay J. Van Bavel, and Elke U. Weber. Stewardship of global collective behavior. *Proceedings of the National Academy of Sciences*, 118(27):e2025764118, 2021. doi: 10.1073/pnas.2025764118.

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

[28] Weixu Zhu, Sinan Oğuz, Mary Katherine Heinrich, Michael Allwright, Mostafa Wahby, Anders Lyhne Christensen, Emanuele Garone, and Marco Dorigo. Self-organizing nervous systems for robot swarms. *Science Robotics*, 9(96):eadl5161, 2024.

[29] Carlo Pinciroli, Mohamed S Talamali, Andreagiovanni Reina, James AR Marshall, and Vito Trianni. Simulating Kilobots within ARGoS: Models and experimental validation. In *International Conference on Swarm Intelligence*, pages 176–187. Springer, 2018.

[30] Fredrik Jansson, Matthew Hartley, Martin Hinsch, Ivica Slavkov, Noemí Carranza, Tjelvar SG Olsson, Roland M Dries, Johanna H Grönqvist, Athanasius FM Marée, James Sharpe, et al. Kilombo: a Kilobot simulator to enable effective research in swarm robotics. *arXiv preprint arXiv:1511.04285*, 2015.

[31] Stefanie Warnat-Herresthal, Hartmut Schultze, Kshitij L. Shastry, Sathyanarayanan Manamohan, Saikat Mukherjee, Vaishnavi Garg, Ramu Sarveswara, Kristian Händler, Peter Pickkers, N. Ahmad Aziz, Sissy Ktena, Florian Tran, Michael Bitzer, Stefan Wuchty, Zeeshan Ashraf, Thomas Flerlage, Evangelos J. Giamarellos-Bourboulis, John P. A. Ioannidis, Khalid Fakhro, Habiba Alsafar, Jesmond Dalli, Gautam Adhikary, Hamish E. Scott, Joachim L. Schultze, and Mihai G. Netea. Swarm learning for decentralized and confidential clinical machine learning. *Nature*, 594(7862):265–270, 2021. doi: 10.1038/s41586-021-03583-3.

[32] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on Large Language Model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.

[33] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22, 2023.

[34] Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 6(3), 2023.

[35] Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. MetaGPT: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 3(4):6, 2023.

[36] Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, Khaled Saab, Dan Popovici, Jacob Blum, Fan Zhang, Katherine Chou, Avinatan Hassidim, Burak Gokturk, Amin Vahdat, Pushmeet Kohli, Yossi Matias, Andrew Carroll, Kavita Kulkarni, Nenad Tomasev, Yuan Guan, Vikram Dhillon, Eeshit Dhaval Vaishnav, Byron Lee, Tiago R D Costa, José R Penadés, Gary Peltz, Yunhan Xu, Annalisa Pawlosky, Alan Karthikesalingam, and Vivek Natarajan. Towards an AI co-scientist. *arXiv preprint arXiv:2502.18864*, 2025.

[37] Daniil A Boiko, Robert MacKnight, and Gabe Gomes. Emergent autonomous scientific research capabilities of large language models, 2023. *arXiv preprint arXiv:2304.05332*, 2023.

[38] Altera AL, Andrew Ahn, Nic Becker, Stephanie Carroll, Nico Christie, Manuel Cortes, Arda Demirci, Melissa Du, Frankie Li, Shuying Luo, et al. Project Sid: Many-agent simulations toward AI civilization. *arXiv preprint arXiv:2411.00114*, 2024.

[39] Ziyi Yang, Zaibin Zhang, Zirui Zheng, Yuxian Jiang, Ziyue Gan, Zhiyu Wang, Zijian Ling, Jinsong Chen, Martz Ma, Bowen Dong, et al. Oasis: Open agents social interaction simulations on one million agents. *arXiv preprint arXiv:2411.11581*, 2024.

[40] Yoichi Ishibashi and Yoshimasa Nishimura. Self-organized agents: A LLM multi-agent framework toward ultra large-scale code generation and optimization. *arXiv preprint arXiv:2404.02183*, 2024.

[41] Huao Li, Yu Quan Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Michael Lewis, and Katia Sycara. Theory of mind for multi-agent collaboration via large language models. *arXiv preprint arXiv:2310.10701*, 2023.

[42] Peihan Li, Zijian An, Shams Abrar, and Lifeng Zhou. Large Language Models for multi-robot systems: A survey. *arXiv preprint arXiv:2502.03814*, 2025.

[43] Mohit Sharma, Simone Baldi, and Tansel Yucelen. Low-distortion information propagation with noise suppression in swarm networks. *Proceedings of the National Academy of Sciences*, 120(11):e2219948120, 2023. doi: 10.1073/pnas.2219948120.

[44] Guande Wu, Chen Zhao, Claudio Silva, and He He. Your co-workers matter: Evaluating collaborative capabilities of language models in Blocks World. *arXiv preprint arXiv:2404.00246*, 2024.

[45] Peihan Li, Vishnu Menon, Bhavanaraj Gudiguntla, Daniel Ting, and Lifeng Zhou. Challenges faced by Large Language Models in solving multi-agent flocking. *arXiv preprint arXiv:2404.04752*, 2024.

[46] Shyam Sundar Kannan, Vishnunandan LN Venkatesh, and Byung-Cheol Min. Smart-LLM: Smart multi-agent robot task planning using large language models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12140–12147. IEEE, 2024.

[47] Bangguo Yu, Hamidreza Kasaei, and Ming Cao. Co-NavGPT: Multi-robot cooperative visual semantic navigation using large language models. *arXiv preprint arXiv:2310.05719*, 2023.

[48] Xudong Guo, Kaixuan Huang, Jiale Liu, et al. Embodied LLM agents learn to cooperate in organized teams. *arXiv preprint arXiv:2403.12482*, 2024.

[49] Xiaopan Zhang, Hao Qin, Fuquan Wang, et al. LaMMA-P: Generalizable multi-agent long-horizon task allocation and planning with LM-driven PDDL planner. *arXiv preprint arXiv:2403.06940*, 2024.

[50] Zhao Mandi, Shreeya Jain, and Shuran Song. Roco: Dialectic multi-robot collaboration with large language models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 286–299. IEEE, 2024.

[51] Junting Chen, Checheng Yu, Xunzhe Zhou, Tianqi Xu, et al. EMOS: EMBODIMENT-AWARE heterogeneous multi-robot operating system with LLM agents. *arXiv preprint arXiv:2405.19012*, 2024.

[52] Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B. Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[53] Hongxin Zhang, Zeyuan Wang, Qiushi Lyu, et al. COMBO: Compositional world models for embodied multi-agent cooperation. *arXiv preprint arXiv:2402.15000*, 2024.

[54] Kunal Garg, Jacob Arkin, Songyuan Zhang, Nicholas Roy, and Chuchu Fan. Large language models to the rescue: Deadlock resolution in multi-robot systems. *arXiv preprint arXiv:2404.14293*, 2024.

[55] DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[56] Huajian Xin, Z. Z. Ren, Junxiao Song, Zhihong Shao, Wanjia Zhao, Haocheng Wang, Bo Liu, Liyue Zhang, Xuan Lu, Qiushi Du, Wenjun Gao, Qihao Zhu, Dejian Yang, Zhibin Gou, Z. F. Wu, Fuli Luo, and Chong Ruan. Deepseek-prover-v1.5: Harnessing proof assistant feedback for reinforcement learning and monte-carlo tree search. *arXiv preprint arXiv:2408.08152*, 2024.

[57] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

[58] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL `https://aclanthology.org/D19-1410`.

[59] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. "O'Reilly Media, Inc.", 2009.

A Physics Simulation Details

The SwarmBench simulation employs a sophisticated discrete physics engine to govern interactions between agents and objects within the 2D grid world. This engine is designed to resolve complex multi-body pushing scenarios, ensuring that collective actions, such as those required in the `Transport` task, are subject to consistent and non-trivial physical laws.

A.1 Core Physical Entities and Properties

Two primary constructs define physical entities:

- `Mesh`: Represents a discrete physical object on the grid. Each `Mesh` has:
  - `pos`: Its global top-left coordinate (i, j).
  - `shape`: A 2D array defining its footprint (e.g., a 1×1 square for an agent, or a 1×4 rectangle for a large obstacle).
  - `static`: A boolean indicating if the object is immovable (e.g., walls `'W'`).
  - `mass` (m): Resistance to motion, calculated as $m = \lfloor \sqrt{\text{area}} \rfloor$, where area is the number of non-empty cells in its `shape`. For a standard 1×1 agent, area is 1, thus mass $m = 1$.
- `Node`: A computational representation used during physics resolution. A `Node` can represent a single `Mesh` or, crucially, an aggregate of `Meshes` that form a Strongly Connected Component (SCC) in the interaction graph (see below). Each `Node` aggregates:
  - Total `mass` and `static` status of the `Mesh`(es) it represents.
  - Net external `force` applied to it by agents or other `Nodes`.

Agents are a specific type of `Mesh` with mass $m = 1$. When an agent performs a movement action (e.g., `UP`, `RIGHT`), it attempts to apply a directed force, typically of magnitude $F = 2$, to an adjacent entity or into empty space.

A.2 Interaction Resolution via SCCs and ILP

The simulation resolves all potential movements and pushes within a single time step through a multi-stage process:

1. **Contact Graph Construction**: The engine identifies all `Mesh` objects that are adjacent and could potentially exert force on one another based on intended agent actions or ongoing pushes. This forms a directed graph where an edge $v \to u$ indicates that `Mesh` v could potentially push `Mesh` u.
2. **Strongly Connected Component (SCC) Reduction**: Tarjan's algorithm is applied to the contact graph to identify all SCCs. An SCC represents a group of `Meshes` that are mutually pushing each other or form a rigid cluster that must move as one unit (or not at all). Each SCC is collapsed into a single aggregate `Node`. `Meshes` not part of any cycle become individual `Nodes`. This process transforms the potentially cyclic contact graph into a Directed Acyclic Graph (DAG) of `Nodes`, representing the pathways of force transmission. The mass and applied forces for an aggregate `Node` are summed from its constituent `Meshes`.
3. **Integer Linear Program (ILP) Formulation and Solution**: The core of the physics resolution is an ILP problem formulated and solved using the `PuLP` library.
   - **Variables**: Binary variables x_v indicate if `Node` v moves; continuous variables represent net forces on nodes and forces transmitted between connected nodes in the DAG.
   - **Objective**: Maximize $\sum x_v$ — i.e., maximize the number of (aggregate) `Nodes` that are successfully moved.
   - **Key Constraints**:
     - *Movement Condition*: A `Node` v can only move ($x_v = 1$) if the total net force $F_{\text{net},v}$ acting on it in the direction of potential movement is greater than or equal to its total mass m_v (i.e., $F_{\text{net},v} \geq m_v$).
     - *Force Transmission*: Force is transmitted along the DAG. A `Node` v can only exert force on its children in the DAG if it itself moves ($x_v = 1$) and has sufficient "leftover" force ($F_{\text{net},v} - m_v$).

- *Static Objects*: `Nodes` marked as `static` (e.g., containing walls) are constrained such that $x_v = 0$.
- *Grid Boundaries*: Movement is implicitly constrained by grid boundaries and collisions with other static objects, handled by the graph construction.

The ILP solver finds the optimal set of `Nodes` that can move simultaneously while satisfying all physical constraints.

4. **Position Update**: The global positions of the `Meshes` belonging to the `Nodes` determined to be movable by the ILP solution are updated on the simulation grid.

This physics model, particularly the SCC reduction and ILP-based resolution, allows SwarmBench to simulate complex, emergent physical interactions that require genuine coordination, such as multiple agents cooperatively pushing a heavy object that no single agent could move alone.

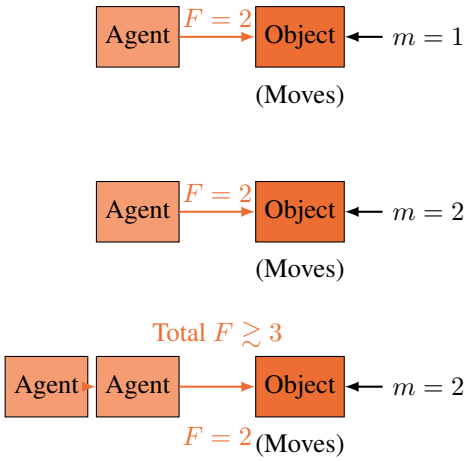


Figure S.1: Illustration of the core physics principle: an applied force (F) versus object mass (m). An agent applies a standard force (typically $F = 2$). An object moves if the net applied force is greater than or equal to its mass ($F \geq m$). **Top:** A single agent ($F = 2$) pushes an object of $m = 1$. Since $2 \geq 1$, the object moves. **Middle:** A single agent ($F = 2$) pushes an object of $m = 2$. Since $2 \geq 2$, the object moves. **Bottom:** Two agents cooperatively push. Agent 3b pushes Agent 3a, which in turn pushes the object. The effective combined force (e.g., $F_{\text{total}} \gtrsim 3$ after accounting for Agent 3a's own mass if it also moves) is applied to the object of $m = 2$. Since $F_{\text{total}} \geq 2$, the object moves. The ILP solver handles the precise calculation of force transmission and movement thresholds in such multi-body scenarios based on the DAG of interactions.

B SwarmBench System and Protocol Details

This appendix provides a detailed description of the SwarmBench environment, agent capabilities, and the evaluation protocol used in our experiments, complementing Section 3 of the main text.

B.1 Environment Details

SwarmBench utilizes a simulation environment based on a 2D grid world where multiple agents (N agents), controlled by LLMs, operate and interact. The adoption of a 2D grid world, while an abstraction, is a deliberate design choice aligned with foundational AI benchmarking practices (e.g., SnakeBench [23] and ARC-AGI [24]). This facilitates a focused investigation of core coordination dynamics while maintaining tractable complexity for initial explorations. This environment itself is designed as a customizable and scalable physical system, where mechanical interactions such as forces and multi-body dynamics (further detailed in Appendix A) are explicitly modeled.

The simulation proceeds in discrete time steps (rounds, $t = 1, \ldots, T$). In each round, all agents perceive their local environment (including messages from the previous round) simultaneously and decide upon their next action and potential message based on the state at the beginning of the round. Environment updates, including agent movement and object interactions, occur only after all agents have committed to their actions for that round. Interactions between agents and objects, particularly pushing and collision resolution, are governed by this discrete physics simulation that handles complex multi-body dynamics, ensuring that the mechanical properties of the system are consistently applied.

The benchmark includes several core multi-agent coordination tasks designed to probe different facets of emergent swarm behavior (visualized in Figure 1 in the main text and detailed with examples in Appendix D):

- **Pursuit:** Agents (e.g., `'0'-'9'`) must collaboratively track and corner a faster-moving prey (`'P'`). Tests coordination for containment, potentially aided by communication.
- **Synchronization:** Agents aim to synchronize an internal binary state (`'Number'` vs. `'$Number'`) across the swarm and collectively alternate this state via a `SWITCH` action. Assesses consensus formation leveraging local cues and communication.
- **Foraging:** Agents navigate an environment with walls (`'W'`) to find a food source (`'F'`), transport it to a nest (`'N'`), changing appearance (`'Number'` to `'$Number'`) when carrying. Evaluates exploration, pathfinding, and potential communication-driven task allocation.
- **Flocking:** Agents must move as a cohesive group, maintaining alignment and separation while potentially navigating towards a target or avoiding obstacles. Tests emergent formation control and coordinated movement.
- **Transport:** Multiple agents must cooperate to push a large object (`'B'`) towards a designated goal area. Tests coordinated force application and navigation around obstacles.

The environment framework supports additional tasks (e.g., Obstacle Pushing, Shape Formation) and is extensible. Interactions follow simplified physics rules detailed in Appendix A. Environment instances, including initial agent positions, object placements, and potentially other environmental features, are procedurally generated based on a random `seed`. To ensure robust evaluation and prevent overfitting to specific scenarios, each simulation run across different models or trials utilizes a distinct seed. This presents varied initial conditions and environmental layouts, promoting the development of generalizable coordination strategies.

B.2 Agent Perception, Action, and Communication Details

Consistent with the goal of studying emergent behavior from local information, agents operate with significantly restricted perception. The primary input is an egocentric $k \times k$ grid view (e.g., 5×5 in our main experiments) centered on the agent at position $\mathbf{x}_{i,t} \in \mathbb{R}^2$. This view displays local entities using symbols: the agent itself (`'Y'`), other agents (by ID, e.g., `'1'/'$1'`), walls (`'W'`), obstacles (`'B'`), empty space (`'.'`), off-map markers (`'*'`), and task-specific objects (`'P'`, `'N'`, `'F'`). The view includes global coordinate labels.

The full observation packet provided to the LLM includes:

- The local $k \times k$ grid view.
- The agent's global coordinates $\mathbf{x}_{i,t}$.
- Task-specific status (e.g., `carrying_food`).
- Messages received from other agents in the previous round ($t-1$). Messages are received only from agents within the sender's perception radius at time $t-1$.
- The task description and current progress indicators (e.g., `score`).
- A limited history of the agent's own recent observations and actions (e.g., last `memory=5` rounds).

The detailed structure and content of the prompt given to the LLM are provided in Appendix C.

Based on this observation, the agent's LLM must decide on two outputs for round t:

1. A primary action $A_{i,t}$ chosen from a set $\mathcal{A}$ typically including basic movements (`UP`, `DOWN`, `LEFT`, `RIGHT`, `STAY`). Movement actions correspond to an agent attempting to apply a directed force (default magnitude = 2). Agents and objects possess inherent weight (referred to as 'mass' in the simulation, default agent mass = 1 calculated from a 1x1 size). Movement or pushing only occurs if the net applied force overcomes the resistance (mass) of the target object(s), considering potentially complex chain reactions resolved by the physics engine (see Appendix A). Task-specific actions (e.g., `SWITCH`, `PICKUP`, `DROP`) are also included.
2. A message $M_{i,t}$ (a string, potentially empty) intended for local broadcast via the `MSG` action.

The message $M_{i,t}$ (if non-empty) is broadcast locally and anonymously to agents within the sender's perception radius, becoming part of their observation packet in the next round ($t+1$). Messages are subject to a character limit (e.g., 120 characters). This setup compels reliance on interpreting local visual cues and utilizing the constrained communication channel for effective coordination.

B.3 Evaluation Protocol Details

We define a standardized protocol focusing on zero-shot LLM evaluation. Each agent i is controlled by an independent LLM instance. In round t, the agent receives its full observation packet (including received messages from $t-1$), formulates a prompt containing this information (see Appendix C), and queries the LLM. Each query is stateless regarding the LLM's internal conversational context; persistence is managed via the prompt's explicit inclusion of observation history and received messages.

The LLM response is parsed to extract the intended primary action $A_{i,t} \in \mathcal{A}$ and the message content $M_{i,t}$. An episode ends upon task success criteria being met or reaching a maximum round limit (`max_round`).

SwarmBench is model-agnostic. Our experiments (Section 4) utilize several contemporary closed-source (API-based) and open-source LLMs, evaluated without task-specific fine-tuning to assess their inherent zero-shot coordination potential derived from pre-training.

C Prompt Design

The following tcolorbox shows the exact structure and content of the prompt string generated by the 'gen_prompt' function and provided to each LLM agent in SwarmBench at each decision step. Placeholders within curly braces (e.g., {name}, {task_desc}, {view_str}) are dynamically filled with actual simulation data during runtime.

SwarmBench Agent Prompt Template

```
"""You are Agent {name}, operating in a multi-agent environment.  Your goal is
to complete the task through exploration and collaboration.

Task description:
{task_desc}

Round:  {round_num}

Your recent {self.memory}-step vision (not the entire map):
{view_str}

Your current observation:
{level_obs_str}

Message you received:
{messages_str}

Your action history:
{history_str}

Symbol legend:
- Number:  An agent whose id is this number (do not mistake column no.  and
line no.  as agent id).
- Y: Yourself.  Others see you as your id instead of "Y".
- W: Wall.
- B: Pushable obstacle (requires at least 5 agents pushing in the same
direction).
- .:  Empty space (you can move to this area).
- *:  Area outside the map.
And other symbols given in task description (if any).

Available actions:
1.  UP: Move up
2.  DOWN: Move down
3.  LEFT: Move left
4.  RIGHT: Move right
5.  STAY: Stay in place
6.  MSG: Send a message
And other actions given in task description (if any).

Physics rules:
1.  Your own weight is 1, and you can exert a force of up to 2.
2.  An object (including yourself) can only be pushed if the total force in
one direction is greater than or equal to its weight.
3.  Static objects like W (walls) cannot be pushed; only B can be pushed.
4.  Force can be transmitted, but only between directly adjacent objects.
That means, if an agent is applying force in a direction, you can push that
agent from behind to help.
5.  Only pushing is allowed - there is no pulling or lateral dragging.  In
other words, to push an object to the right, you must be on its left side and
take the RIGHT action to apply force.

Message rules:
```

```
1.  A message is a string including things you want to tell other agents.
2.  Your message can be received by all agents within your view, and you can
receive messages from all agents within your view.
3.  Messages are broadcast-based.  The source of a message is anonymous.
4.  Write only what's necessary in your message.  Avoid any ambiguity in your
message.
5.  Messages is capped to no more than 120 characters, exceeding part will be
replaced by "...".

Other rules:
1.  Coordinates are represented as (i, j), where i is the row index and j is
the column index.  Your 5x5 vision uses global coordinates, so please use
global coordinates.
2.  The direction of increasing i is downward, and increasing j is to the
right.
3.  Objects that are completely outside the map (marked with "*") will be
removed.

Please think carefully and choose your next action.  You will need to
collaborate with other agents to successfully complete the task.

Your response should include:
1.  Analysis of the current situation
2.  Your decision and reasoning
3.  The message to be left (if any)

End your response clearly with your chosen action:  "ACTION: [YOUR_ACTION]"
and/or "MSG: [Your message (no line breaks).]"
"""
```

Example SwarmBench Prompt (Agent_5, Round 62, Transport Task)

```
You are Agent Agent_5, operating in a multi-agent environment.  Your goal is
to complete the task through exploration and collaboration.

Task description:
The boundary of the map is surrounded by walls (denoted as W), with a gap
leading to the outside of the map (denoted as '*').  The gap is blocked by an
obstacle (denoted as B).
The goal is to first locate the obstacle (B), then have five robots
simultaneously push it through the exit, and finally escape to the outside
of the map (denoted as '*').

Round:  62

Your recent 5-step vision (not the entire map):
Current Step:
     1   2   3   4   5
  3  1   B   B   B   B
  4  9   8   0   11  10
  5  .   .   Y   .   7
  6  .   .   .   .   .
  7  .   .   .   .   .

1 Steps Before:
     1   2   3   4   5
  3  1   B   B   B   B
  4  .   9   8   0   11
  5  .   .   Y   .   7
  6  .   .   .   .   .
  7  .   .   .   .   .
```

```
2 Steps Before:
     1   2   3   4   5
  3  1   B   B   B   B
  4  .   9   8   0   11
  5  .   .   Y   .   7
  6  .   .   .   .   .
  7  .   .   .   .   .

3 Steps Before:
     1   2   3   4   5
  3  1   B   B   B   B
  4  .   9   8   0   11
  5  .   .   Y   .   7
  6  .   .   .   .   .
  7  .   .   .   .   .

4 Steps Before:
     1   2   3   4   5
  3  1   B   B   B   B
  4  .   9   8   0   11
  5  .   .   Y   .   7
  6  .   .   .   .   .
  7  .   .   .   .   .


Your current observation:


Message you received:
Message:  "At (5,3), moving UP to (4,3) for LEFT push on B at (3,3).  Ready
for 5-force."
Message:  "Moving to (4,5) to join LEFT push on B at (3,5).  Ready for 5-force
push."
Message:  "Moving RIGHT to (4,3) then (4,4) for LEFT push on B. Ready to join
force."
Message:  "At (4,5) pushing LEFT on B at (3,5).  5-force confirmed.  Executing
now."


Your action history:
Round 58:  Action:  UP, Message:  "Moving to (4,3) then (3,4) for LEFT push on
B. Need 4 more at (3,4)/(4,4)."
Round 59:  Action:  UP, Message:  "Moving to (4,3) then (3,4) for LEFT push on
B. Need 4 more at (3,4)/(4,4)."
Round 60:  Action:  UP, Message:  ""Moving to (3,3) for LEFT push on B. Need 1
more at (3,3)/(4,3) to complete 5-force.""
Round 61:  Action:  UP, Message:  ""Moving to (4,3) for LEFT push on B at
(3,3).  Ready to join 5-force push.""
Round 62:  Action:  UP, Message:  "At (5,3), moving UP to (4,3) for LEFT push
on B at (3,3).  Ready for 5-force."


Symbol legend:
- Number:  An agent whose id is this number (do not mistake column no.  and
line no.  as agent id).
- Y: Yourself.  Others see you as your id instead of "Y".
- W: Wall.
- B: Pushable obstacle (requires at least 5 agents pushing in the same
direction).
- .:  Empty space (you can move to this area).
- *:  Area outside the map.
And other symbols given in task description (if any).
```

```
Available actions:
1.  UP: Move up
2.  DOWN: Move down
3.  LEFT: Move left
4.  RIGHT: Move right
5.  STAY: Stay in place
6.  MSG: Send a message
And other actions given in task description (if any).

Physics rules:
1.  Your own weight is 1, and you can exert a force of up to 2.
2.  An object (including yourself) can only be pushed if the total force in
one direction is greater than or equal to its weight.
3.  Static objects like W (walls) cannot be pushed; only B can be pushed.
4.  Force can be transmitted, but only between directly adjacent objects.
That means, if an agent is applying force in a direction, you can push that
agent from behind to help.
5.  Only pushing is allowed - there is no pulling or lateral dragging.  In
other words, to push an object to the right, you must be on its left side and
take the RIGHT action to apply force.

Message rules:
1.  A message is a string including things you want to tell other agents.
2.  Your message can be received by all agents within your view, and you can
receive messages from all agents within your view.
3.  Messages are broadcast-based.  The source of a message is anonymous.
4.  Write only what's necessary in your message.  Avoid any ambiguity in your
message.
5.  Messages is capped to no more than 120 characters, exceeding part will be
replaced by "...".

Other rules:
1.  Coordinates are represented as (i, j), where i is the row index and j is
the column index.  Your 5x5 vision uses global coordinates, so please use
global coordinates.
2.  The direction of increasing i is downward, and increasing j is to the
right.
3.  Objects that are completely outside the map (marked with "*") will be
removed.

Please think carefully and choose your next action.  You will need to
collaborate with other agents to successfully complete the task.

Your response should include:
1.  Analysis of the current situation
2.  Your decision and reasoning
3.  The message to be left (if any)

End your response clearly with your chosen action:  "ACTION: [YOUR_ACTION]"
and/or "MSG: [Your message (no line breaks).]"
```

Example LLM Output (JSON)


```
{
    "response": "### Analysis of the Current Situation:\n1. **Obstacle (B)
    Location**: The obstacle B is located at positions (3,2), (3,3), (3,4),
    and (3,5). The exit is likely at one of these positions, but the exact
    exit (*) is not visible in the current vision. However, based on the
    messages, the agents are focusing on pushing B at (3,3) and (3,5) LEFT (
    towards column 1).\n2. **Agent Positions**: \n   - You (Y) are at (5,3). \
    n   - Other agents are at (4,3), (4,5), and possibly others not visible in
     the current vision. \n   - Messages indicate that agents are ready for a
    5-force push on B at (3,3) and (3,5).\n3. **Messages**: \n   - Agents are
    coordinating to push B LEFT at (3,3) and (3,5). \n   - A 5-force push is
    confirmed for B at (3,5), and agents are moving into position for B at
    (3,3).\n4. **Your Role**: You are at (5,3) and can move UP to (4,3) to
    join the push on B at (3,3). Alternatively, you could move to (4,5) to
    assist with the push on B at (3,5), but the messages suggest the push at
    (3,5) is already confirmed with enough agents.\n\n### Decision and
    Reasoning:\n- **Primary Goal**: Assist in the 5-force push on B at (3,3).
    The messages indicate that agents are ready at (4,3) and (4,4) for this
    push. You are at (5,3) and can move UP to (4,3) to join the push.\n- **
    Action**: Move UP to (4,3) to align yourself for the LEFT push on B at
    (3,3). This will contribute to the 5-force needed to move the obstacle.\n-
     **Message**: Confirm your readiness to join the push at (3,3) to ensure
    coordination.\n\n### Message:\n\"At (5,3), moving UP to (4,3) for LEFT
    push on B at (3,3). Ready for 5-force.\"\n\n### Action:\nACTION: UP  \nMSG:
     \"At (5,3), moving UP to (4,3) for LEFT push on B at (3,3). Ready for 5-
    force.\"",
    "action": "UP",
    "message": "\"At (5,3), moving UP to (4,3) for LEFT push on B at (3,3).
    Ready for 5-force.\""
}
```

D Examples

This appendix provides visual examples of the simulation environment for each of the five core SwarmBench tasks. Each figure shows a snapshot from a simulation run, illustrating agent positions, environment features, and the overall task objective.

D.1 Pursuit

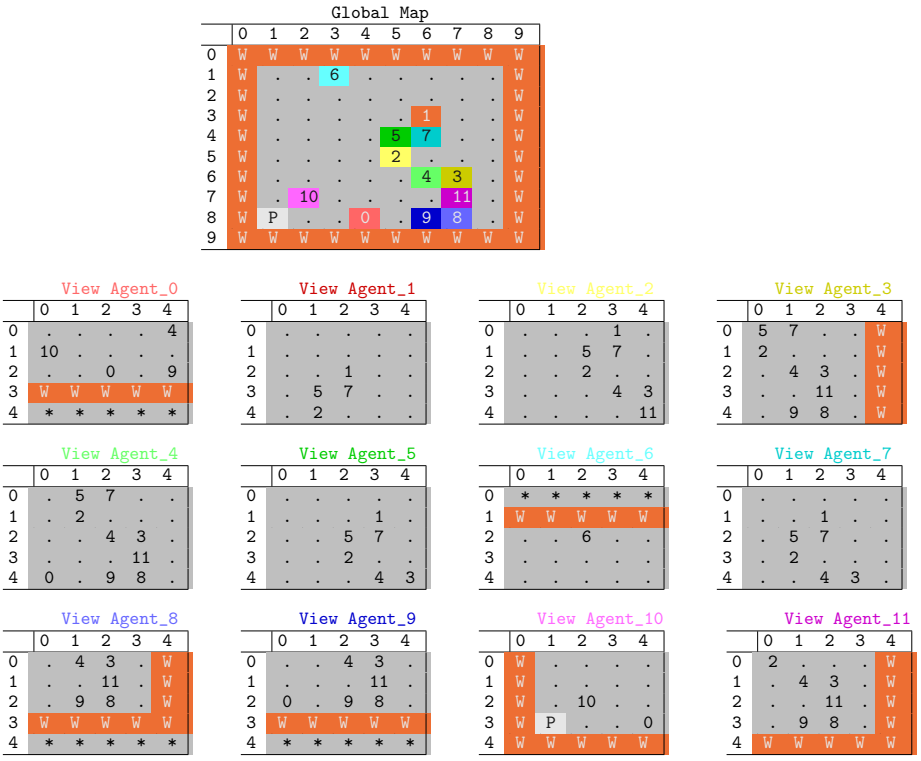
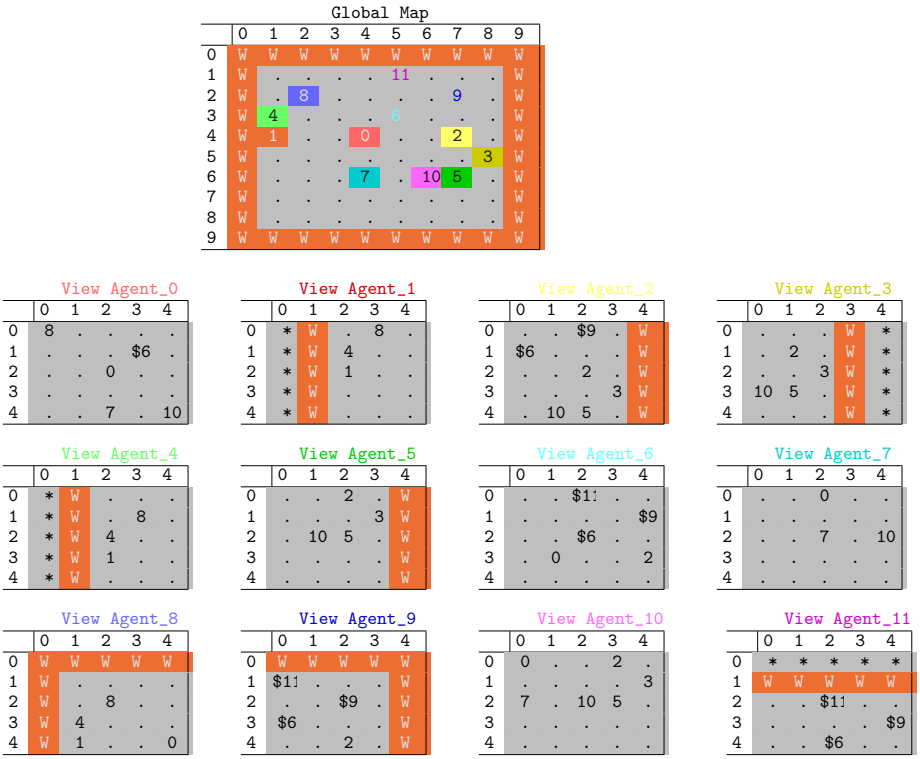


Figure S.2: Example visualization for the `Pursuit` task. Agents (`0-11`) attempt to surround the prey (`P`).

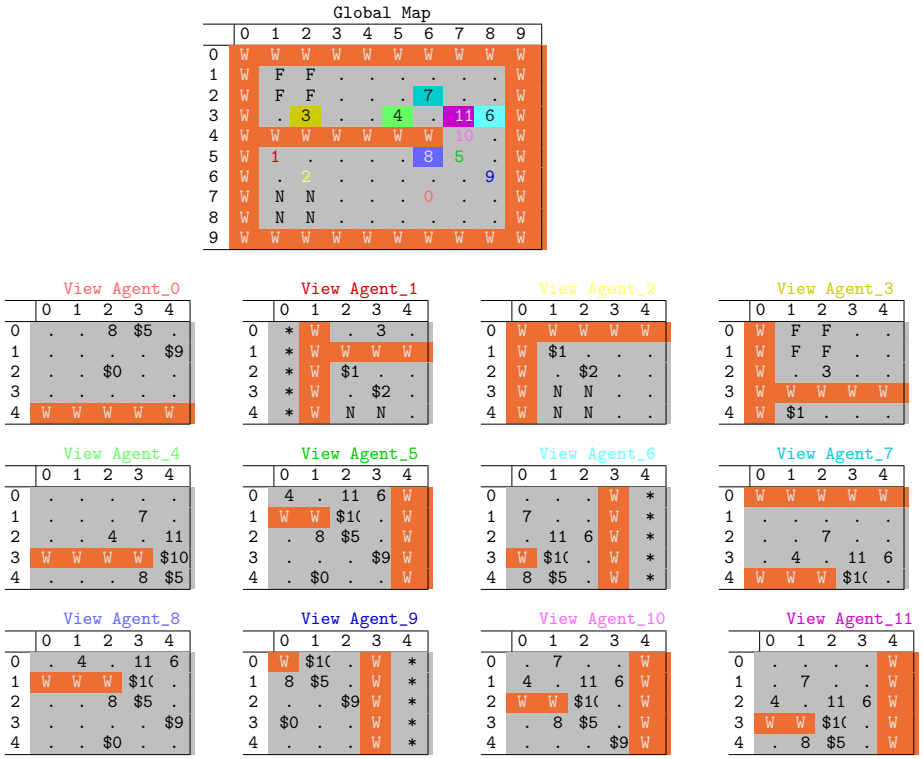
D.2 Synchronization




```
-- Agent Messages --
Agent_0:  0 ON. Target ALL ON! View:  7,8,10 ON; $6$ OFF. Agent 6 SWITCH ON!
Agent 2 (reported OFF) SWITCH ON! Others/Unknowns HOLD...
Agent_1:    Agent 1:  ON. Target ALL ON. See 4 ON, 8 ON. Heard 0 ON. Holding
ON. Echoing Agent 4 \& 8:  Agent 6, if OFF, please SWITCH ...
Agent_2:  2 ON. Target:  ALL ON. See 3, 5, 10 ON; $6, 9$ OFF. Agents 6 and 9,
please SWITCH ON! Let's sync ON. Others hold ON.
Agent_3:       3 ON. Target:  ALL ON. See 2, 5, 10 ALL ON. Good job Agent 2!
Heard from 2 that $6, 9$ are OFF. Agents 6, 9 please SWITCH...
Agent_4:    Agent 4:  ON. Target ALL ON. See 1 ON, 8 ON. Holding ON. Agent 6,
if OFF, please SWITCH ON! All others OFF, switch ON now...
Agent_5:      Target:  ALL ON. 5 ON. See 2, 3, 10 all ON. Good, 2 is ON now!
Agents 6, 9 please SWITCH ON if still OFF. Others HOLD ON.
Agent_6:    6 OFF. Target REMAINS ALL OFF! Saw 0, 2 ON. Agents 0, 2, 3, 5, 7,
8, 10 MUST SWITCH OFF! Others HOLD OFF.
Agent_7:  7 ON. Target ALL ON! See 0, 10 ON. Heard 3,5,8 ON; 2,6 OFF. Agents
2, 6 SWITCH ON! Unknowns (1,4,9,11) ensure ON! HOLD O...
Agent_8:   "Agent 8:  ON. Final Round!  Target ALL ON! View:  0, 1, 4 all ON.
Holding ON. Agent 0 reports 6 OFF, 2 needs ON. Agents 2,...
Agent_9:         9 OFF. Target:  ALL OFF. Agents 0, 2, 5, 10 MUST SWITCH OFF.
Everyone else HOLD OFF.
Agent_10:      10 ON. Target:  ALL ON! Good, Agent 2 is ON now.  Agents 6, 9
please SWITCH ON! Unknowns (1, 4, 11) ensure ON! Others HOLD...
Agent_11:   11 OFF. Target ALL OFF. Confirm 0, 5, 10 SWITCH OFF. Others HOLD
OFF. See $6, $9$ OFF.
```

Figure S.3: Example visualization for the `Synchronization` task. Agents (`Number/$Number`) aim to reach a consensus state.

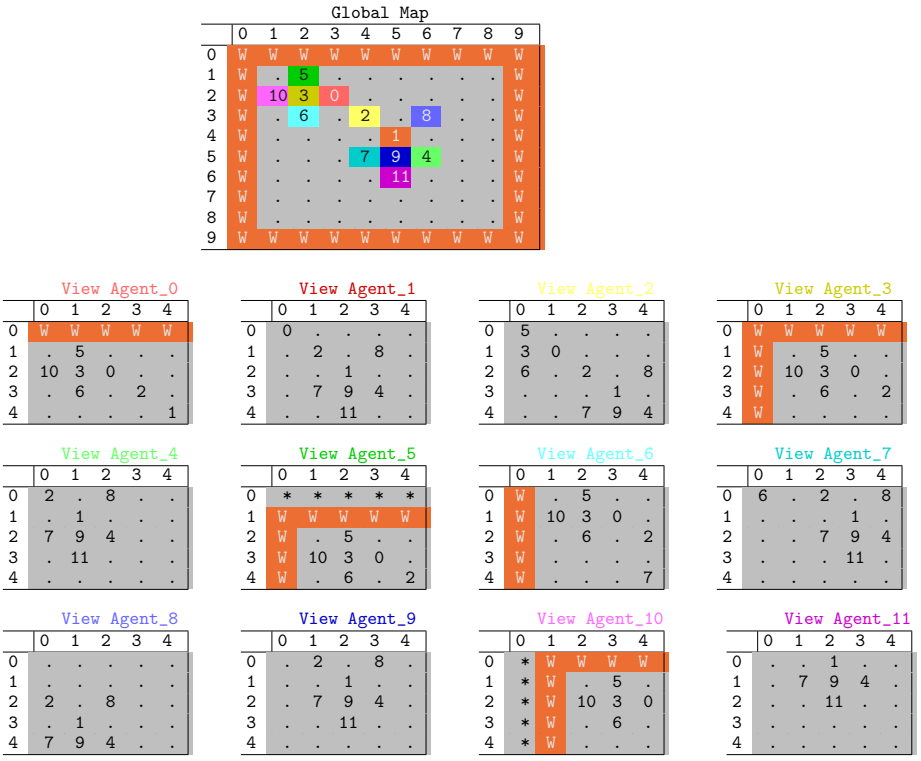
D.3 Foraging




```
-- Agent Messages --
Agent_0:   At (7,7), moving UP to (6,7) towards suspected F near row 5, cols
7-8.  N is West near (7,2).
Agent_1:   At (5,2).  Failed move LEFT (Wall at 5,1).  Moving RIGHT to (5,3)
to explore east route around wall towards F.
Agent_2:  At (6,2), moving RIGHT to (6,3).  Exploring east route around wall
(row 4) to find path North to F.
Agent_3: Agent 3 at (3,2) with food.  Moving RIGHT to (3,3).  Exploring east
path around wall (row 4) towards Nest (7,2).
Agent_4:      Y=Agent 4 at (3,4) with food.  Moving RIGHT to (3,5).  Heading
towards N (~7,2).
Agent_5:       Agent 5 at (5,7).  Moving LEFT to (5,6) to unblock Agent 8 at
F(4,7).  Agent 8, please move DOWN. Agent 10, F is at (4,7)....
Agent_6:     Agent 6 (Y) at (3,8) with food.  Moving DOWN to (4,8).  Heading
towards N near (7,2).
Agent_7:   Y=Agent 7 at (3,5) with food.  Moving UP to (2,5) to bypass row 3
congestion.  Heading East towards N.
Agent_8:         Y=Agent 8 at (5,7) with food.  Moving LEFT to (5,6) towards
N(~7,2).  Making space for $5 at (5,8).
Agent_9:      At (6,7), moving RIGHT to (6,8).  Searching for F, likely near
(5,8) or (4,8).  N is near (7,2).
Agent_10: Moving UP to (3,8) to search for F northwards.  N is near (7,2).
Agent_11:     Y=11 at (3,7) w/ food.  Moving DOWN to (4,7).  Heading towards
N(~7,2).
```

Figure S.4: Example visualization for the `Foraging` task. Agents (`Number`/`$Number`) collect food (`F`) and return it to the nest (`N`).

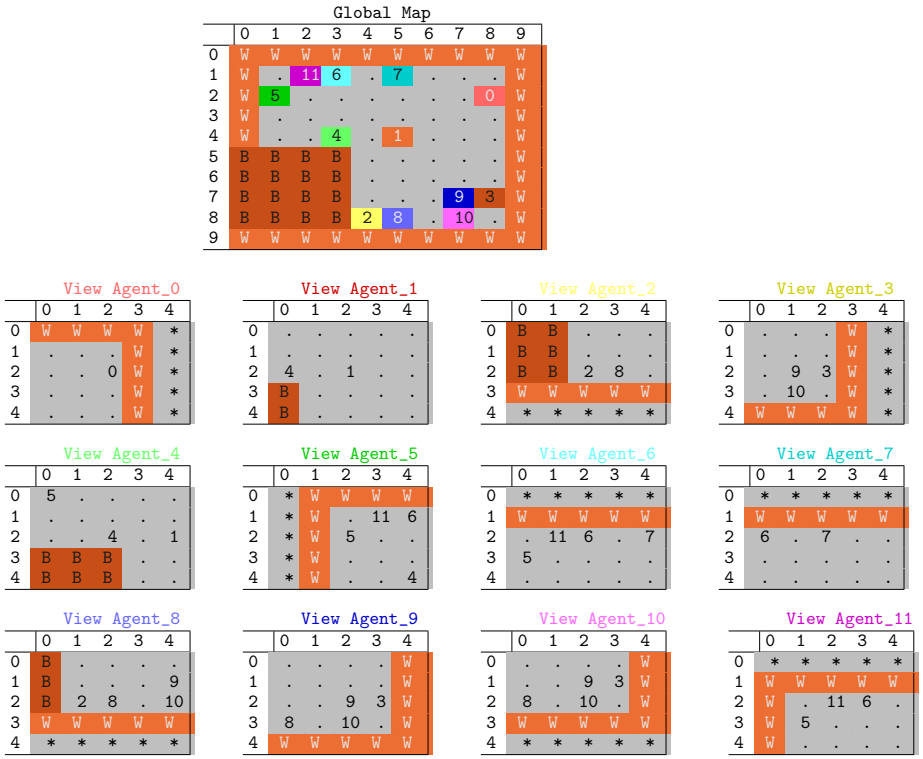
D.4 Flocking




```
-- Agent Messages --
Agent_0: A0 HOLDING (2,3).  Task reported complete.  C1 stable.  HOLDING.
Agent_1:   A1 HOLDING (4,5)[T]. Cross C2@(5,5) [A1,A7,A9,A4,A11] is COMPLETE
and STABLE. Task COMPLETE. HOLDING.
Agent_2:   A2 HOLDING at (3,4).  Received confirmations from C1 \& C2 agents.
Task appears COMPLETE. HOLDING.
Agent_3:     A3 HOLDING (2,2).  Cross 1 [A5(1,2), A10(2,1), A3(2,2), A0(2,3),
A6(3,2)] is STABLE. Awaiting C2 completion or next instr...
Agent_4:      "A4 HOLDING (5,6)[R]. Cross C2@(5,5) [A1,A7,A9,A4,A11] remains
COMPLETE and STABLE. HOLDING position."
Agent_5:       A5 HOLDING at (1,2).  C1 [A5(1,2), A10(2,1), A3(2,2), A0(2,3),
A6(3,2)] stable.  Acknowledging C2 discussion.  HOLDING.
Agent_6:     A6 HOLDING (3,2).  Cross 1 [A5(1,2), A10(2,1), A3(2,2), A0(2,3),
A6(3,2)] remains COMPLETE and STABLE. Holding position.
Agent_7:     A7 HOLDING (5,4)[L]. Cross C2@(5,5) [A1(T), A7(L), A9(C), A4(R),
A11(B)] is COMPLETE and HOLDING. Task complete.  HOLDING...
Agent_8:    A8 HOLDING at (3,6).  Position not part of target shape C1 or C2.
C2 confirmed stable.  Holding position.
Agent_9:    A9 HOLDING (5,5)[C]. Cross C2@(5,5) COMPLETE and STABLE. Holding
position.
Agent_10:   A10 HOLDING (2,1).  Cross 1 [A5(1,2), A10(2,1), A3(2,2), A0(2,3),
A6(3,2)] confirmed complete and stable.  HOLDING.
Agent_11:    A11 HOLDING (6,5)[B]. Cross @(5,5) formed by A1(4,5)T, A7(5,4)L,
A9(5,5)C, A4(5,6)R, A11(6,5)B. TASK COMPLETE. HOLDING.
```

Figure S.5: Example visualization for the `Flocking` task. Agents (`0-11`) attempt to move cohesively.

D.5 Transport




```
-- Agent Messages --
Agent_1: Obstacle B located at (5, 3) and (6, 3).
Agent_2:    Obstacle B at (6,2)-(8,3).  Need 5 agents at row 5 (coords (5,2),
(5,3)) to push DOWN.
Agent_3: Exit at col 10 (rows 5-8).  Wall at col 9.  Seen B?
Agent_4: Obstacle B located at (5,1) to (6,3).
Agent_5: Agent 5 at (2, 2), exploring right for B.
Agent_8:    Obstacle B at (6,3)-(8,3).  Need 5 agents at row 5 to push DOWN.
Moving to push.
Agent_9: Exploring left.  Seen B or *?
```

Figure S.6: Example visualization for the `Transport` task. Agents (`0-11`) coordinate to push a large obstacle (`B`).

E Detailed Group Dynamics Metrics

To quantitatively analyze emergent collective behaviors, we compute metrics primarily based on agent positions $\mathbf{x}_{i,t}$ and their primary actions $A_{i,t}$ (which include movements like `UP`, `DOWN`, `LEFT`, `RIGHT`, inaction `STAY`, and any task-specific primary actions defined for a level). While message content analysis is possible (see Appendix G), the metrics below focus on overt behavioral patterns and spatial configurations.

Behavioral Patterns Characterize the distribution and diversity of primary actions taken by the agents.

- **Action Proportions:** Calculate the overall frequency of specific types of primary actions across all agents and all rounds. For example, the proportion of 'stay' actions is computed as:

$$\texttt{prop\_stay\_actions} = \frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N}\mathbb{I}(A_{i,t} = \texttt{STAY}) \tag{1}$$

  where N is the number of agents, T is the total number of rounds, $A_{i,t}$ is the primary action of agent i at round t, $\mathcal{A}$ is the set of all available primary actions, and $\mathbb{I}(\cdot)$ is the indicator function (1 if the condition is true, 0 otherwise). Proportions for other action subsets (e.g., movement actions $\mathcal{A}_{\text{move}} = \{\texttt{UP}, \texttt{DOWN}, \texttt{LEFT}, \texttt{RIGHT}\}$) are calculated similarly (e.g., `prop_move_actions`).
- **Per-Round Primary Action Entropy (H_t):** Measures the instantaneous diversity (unpredictability) of primary actions within a single round t. It is calculated using the Shannon entropy formula based on the proportion $p_t(a)$ of agents performing action $a \in \mathcal{A}$ in round t:

$$H_t = -\sum_{a\in\mathcal{A}} p_t(a)\log_2 p_t(a) \quad \text{where} \quad p_t(a) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(A_{i,t} = a) \tag{2}$$

  By convention, $0\log_2 0 = 0$. We compute the average over all rounds: $\texttt{avg\_action\_entropy} = \frac{1}{T}\sum_{t=1}^{T} H_t$, and the standard deviation: $\texttt{std\_action\_entropy} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(H_t - \overline{H})^2}$, where $\overline{H}$ is the mean entropy over rounds.
- **Total Primary Action Entropy (H_{total}):** Measures the overall diversity of primary actions used throughout the entire episode, considering the global frequency $p(a)$ of each action $a \in \mathcal{A}$:

$$\texttt{action\_entropy\_total} = -\sum_{a\in\mathcal{A}} p(a)\log_2 p(a) \quad \text{where} \quad p(a) = \frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N}\mathbb{I}(A_{i,t} = a) \tag{3}$$

Movement Coordination Assess the alignment and uniformity of agent actions related to movement or staying put. Let the set of coordination-relevant actions be $\mathcal{A}_{\text{coord}} = \mathcal{A}_{\text{move}} \cup \{\texttt{STAY}\} \subseteq \mathcal{A}$.

- **Per-Round Dominant Action Proportion (D_t):** Measures the proportion of agents performing the single most frequent action within the set $\mathcal{A}_{\text{coord}}$ at round t. Let $N_t^{\text{coord}} = \sum_{i=1}^{N}\mathbb{I}(A_{i,t} \in \mathcal{A}_{\text{coord}})$ be the number of agents performing a coordination-relevant action in round t. If $N_t^{\text{coord}} > 0$:

$$D_t = \max_{a\in\mathcal{A}_{\text{coord}}}\left(\frac{1}{N_t^{\text{coord}}}\sum_{i=1}^{N}\mathbb{I}(A_{i,t} = a)\right) \tag{4}$$

  Otherwise, $D_t = 0$. We compute the average $\texttt{avg\_dominant\_action\_prop} = \frac{1}{T}\sum_{t=1}^{T} D_t$ and the standard deviation $\texttt{std\_dominant\_action\_prop} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(D_t - \overline{D})^2}$ (where $\overline{D}$ is the mean) to measure the average level and temporal variability of behavioral uniformity, respectively.

- **Per-Round Polarization Index (P_t):** Measures the degree of alignment of intended movement vectors $\mathbf{v}(a)$ associated with primary actions in $\mathcal{A}_{\text{coord}}$. We assign standard unit vectors for movement (e.g., $\mathbf{v}(\texttt{UP}) = (0, -1)$, $\mathbf{v}(\texttt{RIGHT}) = (1, 0)$ - assuming grid coordinates where row index increases downwards) and a zero vector for inaction ($\mathbf{v}(\texttt{STAY}) = (0, 0)$). The index is the magnitude of the average movement vector:

$$P_t = \left\| \frac{1}{N} \sum_{i=1}^{N} \mathbf{v}(A_{i,t}) \right\|_2 \quad \text{(where } A_{i,t} \in \mathcal{A}_{\text{coord}}\text{)} \tag{5}$$

  where $\| \cdot \|_2$ denotes the Euclidean norm (vector length). A value near 1 indicates strong alignment, while a value near 0 indicates disordered movement or widespread inaction. We compute the average `avg_polarization_index` and the standard deviation `std_polarization_index`.

In addition to these behavioral metrics, standard spatial metrics such as average pairwise distance (`avg_pairwise_distance`) measuring group dispersion and average centroid speed (`avg_centroid_speed`) measuring collective displacement are computed based on agent positions $\mathbf{x}_{i,t}$ over time using their standard definitions. These combined metrics provide a quantitative basis for analyzing the emergent strategies and collective dynamics discussed in Section 4 and Appendix H.

F Detailed Task Performance Data

Table S.1 provides the detailed numerical results corresponding to the performance overview presented in Figure 3 in the main text (Section 4.1). It shows the mean scores and standard deviations for each evaluated LLM across the five SwarmBench tasks, averaged over 5 simulation runs. Models are ordered by their total score (sum across the five tasks) in descending order.

Table S.1: **Detailed average scores with standard deviations (mean ± std) for various LLMs across five SwarmBench tasks, plus total score.** Tasks: `Pursuit`, `Synchronization`, `Foraging`, `Flocking`, `Transport`. Scores averaged over 5 simulations. Standard deviation shown after ± symbol. Models ordered by Total Score (descending). This data is visualized in Figure 3.

Model	Pursuit	Synchronization	Foraging	Flocking	Transport	Total Score
`gemini-2.0-flash`	8.80 ± 1.60	3.40 ± 2.94	5.80 ± 4.35	10.40 ± 4.72	7.66 ± 2.43	36.06
`claude-3.7-sonnet`	4.40 ± 1.20	12.60 ± 9.62	1.20 ± 1.47	13.60 ± 3.20	4.14 ± 5.07	35.94
`o4-mini`	9.60 ± 0.49	2.80 ± 1.17	4.80 ± 2.64	17.80 ± 1.47	0.16 ± 0.31	35.16
`deepseek-v3`	4.20 ± 2.48	4.00 ± 1.41	2.60 ± 2.06	15.00 ± 1.67	5.84 ± 4.81	31.64
`gpt-4.1`	8.40 ± 1.85	2.80 ± 0.75	3.20 ± 1.94	13.40 ± 2.24	1.53 ± 3.07	29.33
`o3-mini`	3.60 ± 2.06	2.20 ± 1.17	2.60 ± 3.88	13.40 ± 1.74	2.57 ± 3.52	24.37
`gpt-4o`	3.40 ± 1.50	1.80 ± 1.33	1.60 ± 1.85	14.60 ± 2.24	1.38 ± 2.76	22.78
`gpt-4.1-mini`	1.40 ± 0.80	0.60 ± 0.49	1.40 ± 1.02	12.20 ± 2.14	6.30 ± 2.11	21.90
`llama-4-scout`	1.20 ± 0.75	0.20 ± 0.40	1.00 ± 1.55	14.80 ± 2.71	1.15 ± 2.31	18.35
`qwq-32b`	2.20 ± 1.94	1.20 ± 0.98	0.80 ± 0.75	7.00 ± 0.63	7.11 ± 3.67	18.31
`llama-3.1-70b`	1.80 ± 0.40	1.00 ± 1.10	0.00 ± 0.00	13.80 ± 1.17	0.56 ± 1.12	17.16
`deepseek-r1`	1.00 ± 0.63	1.20 ± 1.17	1.00 ± 1.10	8.40 ± 1.96	0.00 ± 0.00	11.60
`claude-3.5-haiku`	0.60 ± 0.49	1.00 ± 0.00	0.00 ± 0.00	3.60 ± 3.38	1.70 ± 2.77	6.90

G Communication Analysis Methodology

To investigate the potential role of explicit communication, facilitated by the `MSG` action, we performed a supplementary analysis focused on the content and patterns of messages exchanged between agents. This analysis utilized standard natural language processing techniques applied to the simulation log data.

This communication analysis was conducted on the complete set of simulation logs across all evaluated models and tasks. The primary goals were to quantify basic communication tendencies (frequency, message length), assess the semantic properties of messages, and identify prevalent keywords.

G.1 Run-Level Communication Metrics

For subsequent correlation with overall task performance (final score, see Appendix I), several metrics characterizing communication within each individual simulation run were computed:

- **Non-Empty Message Frequency (`non_empty_msg_freq_run`):** This metric quantifies the propensity of agents within a run to communicate. It represents the proportion of total agent actions in that run which included the generation of a non-empty message string.
- **Average Non-Empty Message Length (`avg_non_empty_msg_length_run`):** This metric reflects the average verbosity of communication within a run. It is the mean character length calculated over all non-empty messages produced by any agent during that specific simulation run.
- **Average Semantic Similarity (`avg_similarity_run`):** To gauge the overall semantic coherence of communication within a run, message embeddings were generated using the pre-trained Sentence-BERT model `'all-MiniLM-L6-v2'` [58]. This metric represents the average pairwise cosine similarity computed across all embeddings of non-empty messages within that run. Higher values suggest that messages conveyed, on average, more similar semantic content.
- **Semantic Similarity Standard Deviation (`std_similarity_run`):** This metric captures the variability or stability of semantic content communicated throughout a run. It is the standard deviation of the pairwise cosine similarities used to compute the average similarity. Lower values indicate greater semantic consistency among messages, while higher values suggest more fluctuation or divergence in the topics or intents expressed.

G.2 Keyword Analysis

We also performed keyword extraction on the sampled message data to understand the specific terminology used by different models across tasks. Messages were preprocessed (lowercasing, punctuation removal, English stopword removal using NLTK [59]), and the most frequent terms were identified for each model and task combination.

Figure S.7 visually summarizes the results of this analysis, displaying the frequency of the most prominent keywords for each task (grouped by color) across the different LLM models (x-axis). As the figure illustrates, the dominant keywords strongly reflected the specific objectives and entities of each task. For instance, terms like `'push'`, `'left'`, `'obstacle'`, and coordinates (e.g., `'(3,3)'`) were prevalent in the `Transport` task, while `'target'`, `'surround'`, `'corner'`, and agent IDs featured heavily in `Pursuit` messages. Similarly, `'food'`, `'nest'`, `'carry'`, and `'search'` were common in `Foraging`.

This qualitative analysis, visualized in Figure S.7, confirms that the agents' messages frequently contained task-relevant vocabulary. The figure also reveals variations in keyword usage patterns between different LLM models even when performing the same task, suggesting model-specific communication styles or strategies. While the presence of relevant keywords indicates some level of task understanding channeled into communication, their sheer frequency does not directly translate to coordination effectiveness, which, as discussed in the main text (Section 4.2), appeared more strongly linked to emergent physical dynamics and, to a lesser extent, semantic consistency rather than just keyword usage.

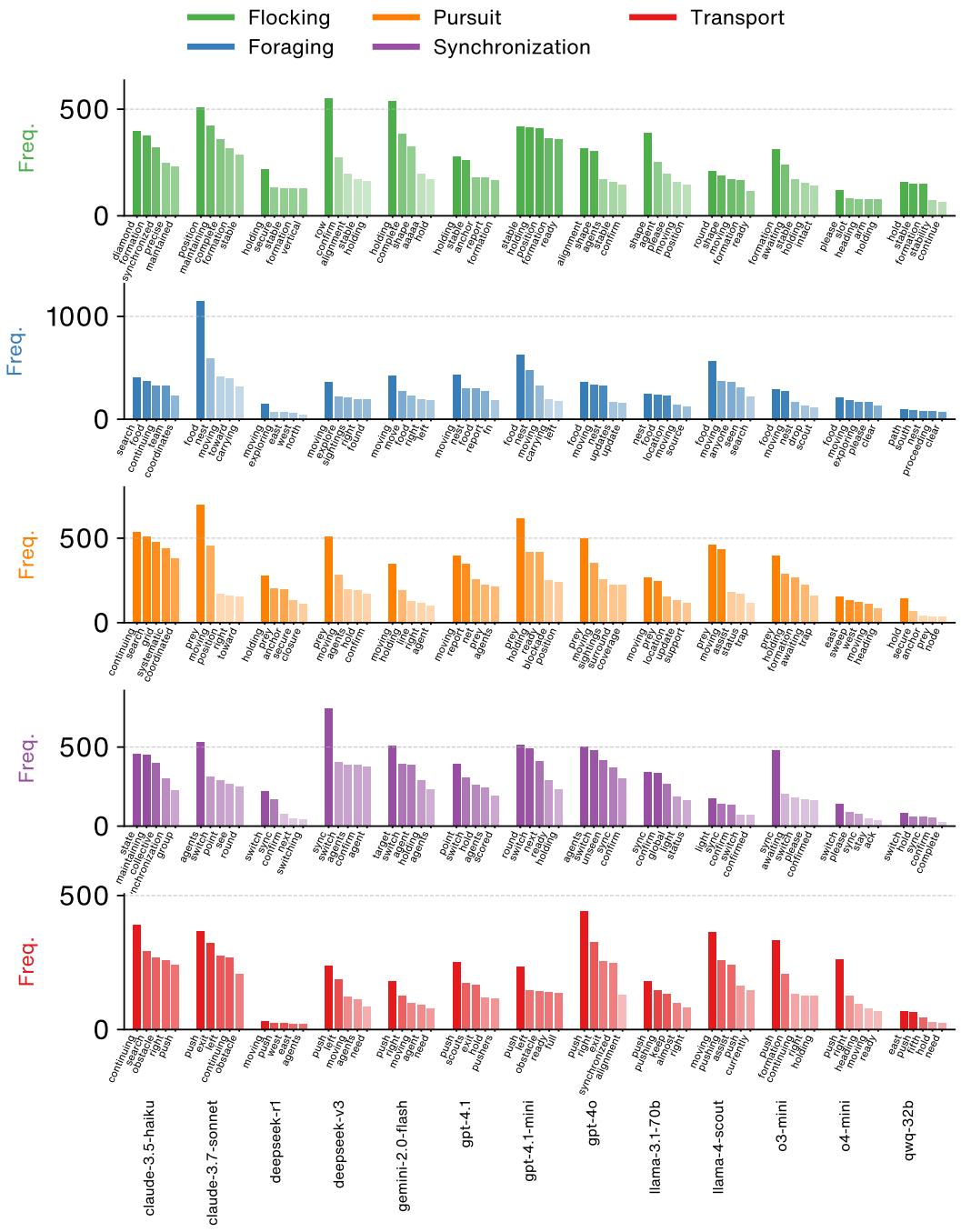


Figure S.7: **Keyword Frequency Analysis from Agent Messages.** Frequency (y-axis) of the top keywords extracted from agent messages, grouped by LLM model (x-axis labels) and task (indicated by color and the vertical plot structure; see legend at top). Message data was preprocessed before frequency counting. The visualization highlights task-specific terminology (e.g., `'push'` in `Transport`, `'food'` in `Foraging`) and reveals variations in keyword usage across different models for the same task.

H Detailed Group Dynamics Correlation and Prediction Model Results

This section provides the detailed quantitative results and visualizations supporting the analysis of emergent group dynamics across all five tasks presented in Section 4.2. The data is aggregated from all 325 simulation runs (across 13 models, 5 tasks, 5 runs each). The metrics used are defined in Appendix E.

H.1 Feature Correlations with Score

Table S.2 lists all primary action-based group dynamics features exhibiting statistically significant Pearson correlations ($p < 0.05$) with the final task score (`score`) across the combined dataset, sorted by the absolute value of the correlation coefficient (r).

Table S.2: **Significant Correlations between Group Dynamics Features and Score (Combined Tasks).** Pearson's r and p-values calculated across all 325 simulation runs. Features are sorted by absolute correlation magnitude. Only significant correlations ($p < 0.05$) are shown. Colors indicate correlation strength/direction.

Feature	r	p-value	Direction
`std_action_entropy`	0.300	< 0.001	Positive
`prop_stay_actions`	0.297	< 0.001	Positive
`std_dominant_action_prop`	0.274	< 0.001	Positive
`avg_polarization_index`	−0.241	< 0.001	Negative
`prop_move_actions`	−0.222	< 0.001	Negative
`num_rounds`	−0.141	0.011	Negative
`avg_action_entropy`	−0.121	0.029	Negative

H.2 Linear Regression Model for Score Prediction

Table S.3 presents the standardized coefficients for the linear regression model built to predict the final task score using ten key primary action-based dynamics features across the combined dataset. The model's overall performance was $R^2 \approx 0.245$ and Mean Squared Error (MSE) ≈ 21.519, using all 325 samples. Features are ranked by the absolute value of their standardized coefficient, indicating their relative importance in the linear model.

Table S.3: **Linear Regression Model Coefficients for Predicting Score (Combined Tasks).** Features ranked by absolute standardized coefficient value. Model $R^2 \approx 0.245$, MSE ≈ 21.519, $N = 325$. Colors indicate effect size/direction.

Feature	Coefficient (Standardized)	Interpretation (Approx. effect of +1 std dev)
`avg_dominant_action_prop`	−5.698	Score decreases by 5.70
`avg_action_entropy`	−4.996	Score decreases by 5.00
`prop_move_actions`	−3.260	Score decreases by 3.26
`action_entropy_total`	2.224	Score increases by 2.22
`std_dominant_action_prop`	1.211	Score increases by 1.21
`std_polarization_index`	−0.555	Score decreases by 0.56
`num_rounds`	−0.450	Score decreases by 0.45
`avg_polarization_index`	−0.360	Score decreases by 0.36
`std_action_entropy`	−0.279	Score decreases by 0.28
`prop_stay_actions`	0.009	Score increases by 0.01

H.3 Visualization of Combined Dynamics

Figures S.8 and S.9 provide visual representations of the correlation analysis performed on the combined dataset across all five tasks.

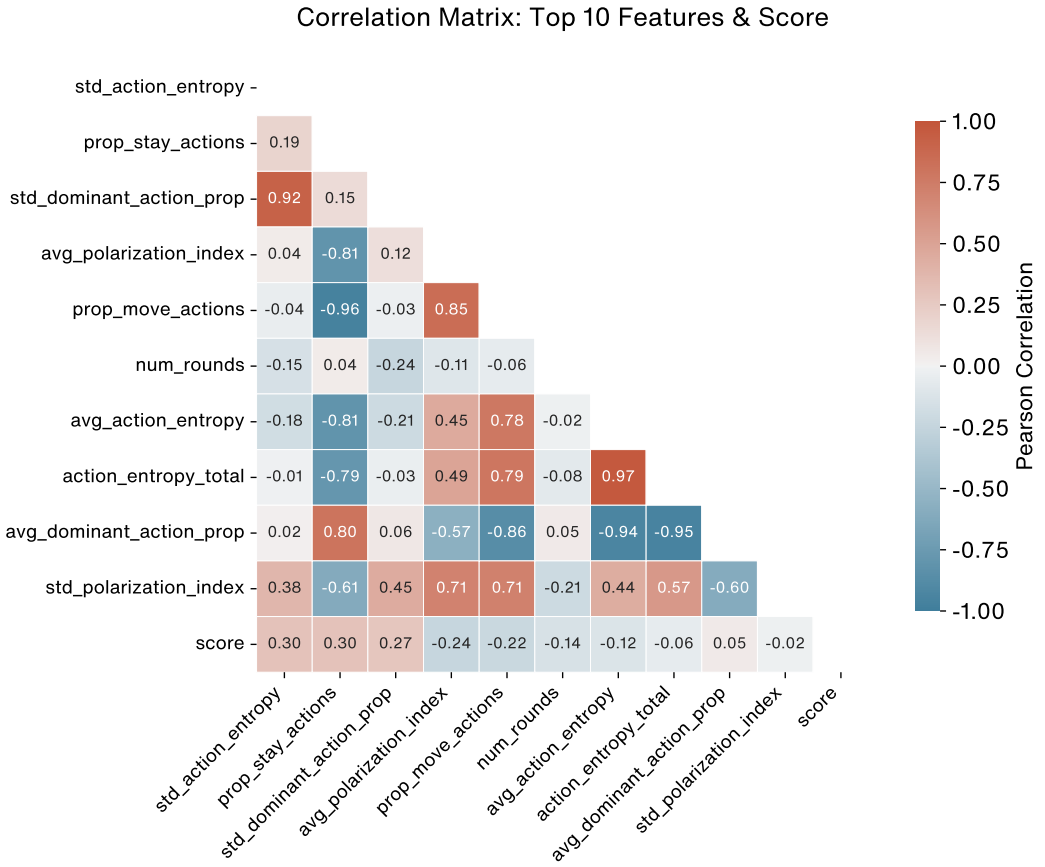


Figure S.8: **Correlation Matrix of Top 10 Numeric Features and Score (Combined Tasks).** This heatmap visualizes the Pearson correlation coefficients (r) between the score and the 10 primary action-based dynamics features analyzed in the combined dataset. The color intensity and hue indicate the strength and direction of the correlation (red: positive, blue: negative). Values are shown within each cell. This provides a visual overview of both feature-score relationships and inter-feature correlations.

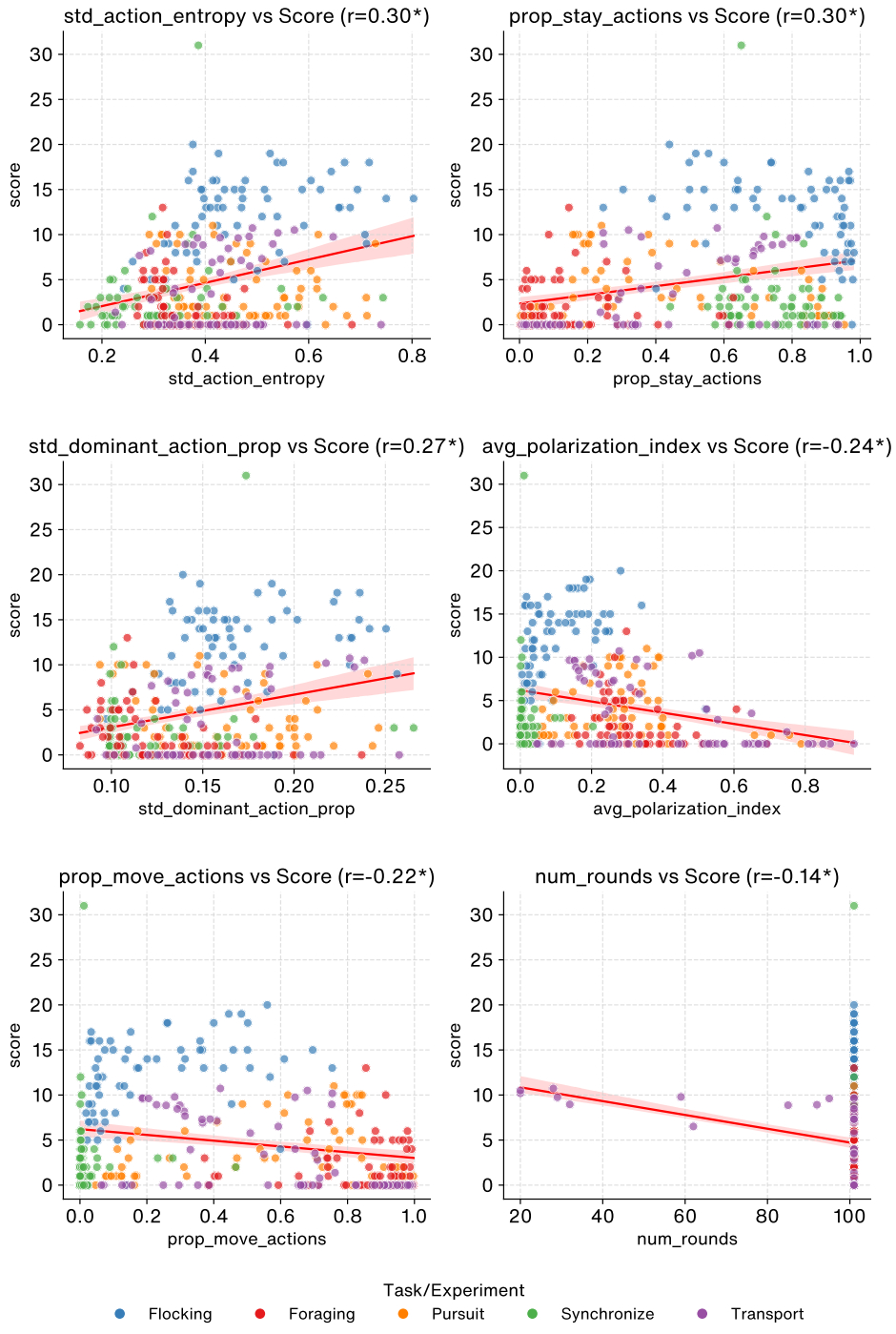


Figure S.9: **Score vs. Top 6 Correlated Features (Combined Tasks, Colored by Task).** These scatter plots show the relationship between the final score and the six group dynamics features exhibiting the strongest absolute Pearson correlation (r) across the combined dataset. Each point represents one simulation run, colored according to its task (`experiment_id`). The red line indicates the overall linear regression fit for the combined data. Plot titles show the feature name and the overall correlation coefficient (r) with score (* indicates $p < 0.05$). These plots help visualize the data distribution, linearity, and potential task-specific patterns underlying the overall correlations.

I Communication Analysis Correlations with Score

This section presents the quantitative correlation results from the supplementary communication analysis, the methodology for which is described in Appendix G. These findings, should be considered in conjunction with the primary dynamics correlations reported in Appendix H.

Table S.4 summarizes the statistically significant Pearson correlations ($p < 0.05$) observed between run-level communication/semantic features and the final task score within the analyzed sample.

Table S.4: **Significant Correlations between Communication/Semantic Features and Score.** Pearson's r and p-values from analysis of approx. 325 simulation runs. Features sorted by absolute $|r|$. Only significant correlations ($p < 0.05$) are listed. Colors indicate effect direction.

Feature	r	p-value	Type / Interpretation
`avg_non_empty_msg_length_run`	0.191	0.0005	Communication (Length +)
`std_similarity_run`	−0.170	0.002	Semantics (Instability -)

Figure S.10 visualizes the relationships between the task score and the two significantly correlated communication/semantic features identified in the sample.

Figure S.10: **Score vs. Significant Communication/Semantic Features.** Scatter plots illustrating the relationship between final task score and (left) average non-empty message length per run (`avg_non_empty_msg_length_run`), and (right) standard deviation of semantic similarity per run (`std_similarity_run`). Red lines depict the overall linear regression trend. Plot titles include the feature name and Pearson correlation coefficient (r) with score (* denotes $p < 0.05$).

As highlighted in Section 5, the magnitude of these correlations associated with communication characteristics is notably lower than those observed for the physical group dynamics metrics (Appendix H, Table S.2). The features `non_empty_msg_freq_run` ($p = 0.77$) and `avg_similarity_run` ($p = 0.91$) did not show statistically significant correlations with the score in this sample analysis.

J Parameter Sensitivity Analysis

We examined how agent performance responds to changes in local perception range (k, the size of the square view) and group size (N, the number of agents), revealing key aspects of decentralized coordination challenges (Figure S.11).

Expanding the field of view from $k = 3$ to $k = 5$ consistently improved outcomes across diverse tasks like `Pursuit`, `Synchronization`, `Foraging`, and `Flocking`. This suggests that a minimal level of environmental awareness is crucial for agents to effectively coordinate, likely enabling better anticipation and response to neighbors' actions. However, further increasing the view to $k = 7$ yielded only marginal gains and was sometimes less effective than $k = 5$, particularly in the `Transport` task which demands precise collective alignment. This plateau, and in some cases like the `Transport` task a performance dip with $k = 7$ compared to $k = 5$, implies a potential trade-off. While more information can be beneficial, an overly broad view might lead to **information overload**, making it harder for the LLM agents to discern critical local cues from a larger, potentially noisier, perceptual field. This could **dilute focus** on immediately relevant neighbors or environmental features crucial for tightly coupled maneuvers, such as the precise alignment needed in `Transport`. The increased cognitive load of processing a larger input space without a corresponding improvement in strategic depth might thus be counterproductive in certain scenarios. The effectiveness of $k = 5$ in our main experiments (Section 4) likely reflects a more optimal balance between sufficient environmental awareness and manageable perceptual complexity for the current LLM architectures in these zero-shot settings.

The influence of group size (N) presented a more complex picture, strongly modulated by task demands. Predictably, performance in `Transport` improved with more agents ($N = 16$ vs $N = 8$), as the task fundamentally relies on accumulating sufficient physical force. Conversely, `Foraging` performance deteriorated as N increased, suggesting that larger groups introduced detrimental effects like congestion or interference near critical locations (nest `'N'`, food `'F'`), outweighing any potential benefits. Intriguingly, `Pursuit` exhibited peak performance at an intermediate size ($N = 12$ compared to $N = 8$ and $N = 16$), hinting that while more agents can help initially encircle a target, too many may hinder coordinated containment through increased complexity and potential self-obstruction. `Flocking` remained relatively robust to changes in N within the tested range.

These varied scaling behaviors highlight a core challenge for LLM-based swarms: managing the increased interaction density and potential for conflicting local decisions in larger groups without centralized control. The sensitivity to both k and N underscores that robust swarm intelligence requires strategies adaptable to varying information availability and group dynamics, motivating evaluation across diverse parametric settings as discussed in Sections 5 and 7.

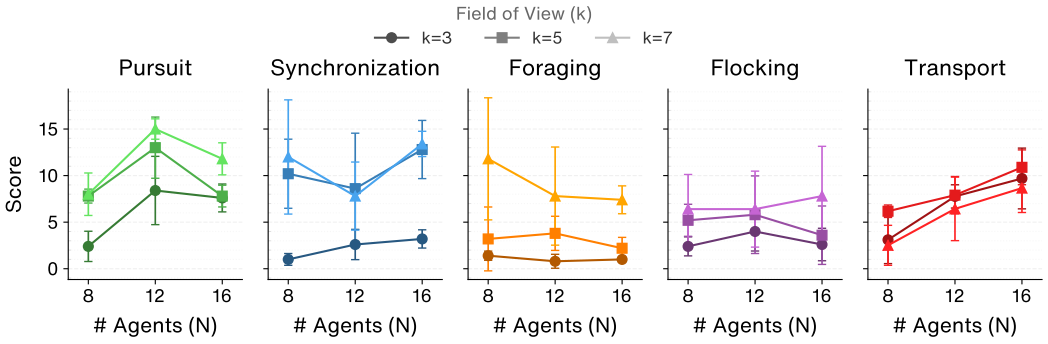


Figure S.11: Parameter Sensitivity Analysis. Performance (Score) across selected tasks varies with the number of agents (N) and field of view size ($k \times k$). Mean scores over runs shown; error bars indicate std. dev. Results suggest optimal parameter ranges can be task-dependent.