

Multi-Agent Reinforcement Learning for Autonomous Large Language Model Collaboration

Warjas Yogi

Department of Electrical Engineering and Computer Science, University of Missouri,
Columbia, MO, USA.
warjas.yogi929@missouri.edu

Urbano Isern

Department of Computer Science, University of New Hampshire, Durham, NH, USA.
isern1986@unh.edu

Geronimo Mocenigo

Department of Computer Science, University of North Texas, Denton, TX, USA.
geronimomocenigo@unt.edu

Milton Jansson

Department of Computer Science, University of Alabama at Birmingham, Birmingham, AL,
USA.
miltonjansson92@uab.edu

Abstract

Large language models have moved rapidly from isolated text generators to components within broader agentic systems, yet the most consequential next step is not simply making individual models more capable. It is enabling many language-model agents to collaborate autonomously under conditions that remain robust, accountable, fair, and sustainable. This paper examines multi-agent reinforcement learning as a system-level framework for autonomous large language model collaboration. It argues that collaboration among language-model agents should be understood as a socio-technical infrastructure problem involving architecture, credit assignment, communication protocols, deployment economics, governance, and policy. The analysis first reviews the shift from single-agent reinforcement learning and language-model assistance to multi-agent language ecologies. It then examines centralized, decentralized, hierarchical, and market-oriented coordination structures, along with their trade-offs in latency, resilience, interpretability, and control. The paper discusses learning dynamics, including non-stationarity, credit assignment, preference alignment, and verbal reinforcement, and considers how these dynamics shape emergent norms and failure modes. It further analyzes infrastructure and sustainability pressures, especially compute intensity, orchestration overhead, observability, and long-term maintenance. Robustness, security, fairness, and governance are treated as first-order design constraints rather than post hoc corrections. Through case illustrations in software development, scientific assistance, public administration, and interactive simulation, the paper shows that autonomous collaboration can improve coverage and adaptability while introducing new accountability gaps. The conclusion recommends layered governance, auditable coordination traces, resource-aware scheduling, and evaluation regimes that measure collective behavior rather than only individual model performance.

Keywords

multi-agent reinforcement learning; large language models; autonomous collaboration; socio-technical systems; governance; robustness; fairness; sustainability; deployment; infrastructure.

1. Introduction

The rapid diffusion of large language models has created a new class of computational actor. These models can summarize, plan, write code, negotiate constraints, and simulate expert dialogue, often with enough fluency to appear as participants in organizational workflows [5]. At the same time, the most difficult problems in real deployments rarely reduce to a single request-response exchange. They involve multiple roles, conflicting objectives, partial information, and sequential decision-making. This has motivated interest in multi-agent reinforcement learning as a way to coordinate collections of language-model agents so that they can divide labor, critique one another, and adapt over time [1], [8]. The resulting systems are not merely ensembles of models. They are distributed socio-technical arrangements in which learning algorithms, communication channels, resource constraints, human oversight, and institutional rules jointly determine outcomes.

Autonomous collaboration among language-model agents raises questions that cannot be answered by model benchmarking alone. A system may achieve higher task success while becoming less predictable, more expensive, or harder to audit. It may improve robustness in one domain while amplifying bias in another. It may reduce human workload in routine cases while creating new supervisory burdens. These trade-offs are structural. They depend on whether coordination is centralized or decentralized, whether credit is assigned through global rewards or local signals, whether agents can form coalitions, and whether deployment is confined to a single organization or spread across jurisdictional boundaries. Multi-agent reinforcement learning offers a vocabulary for these dynamics, but it must be integrated with infrastructure analysis, governance design, and policy evaluation [22], [23].

This paper develops a system-level account of multi-agent reinforcement learning for autonomous large language model collaboration. It does not present a new algorithm or a narrow empirical result. Instead, it synthesizes established work on reinforcement learning, multi-agent coordination, language-model agency, and responsible artificial intelligence into a framework for analyzing architecture, learning, deployment, robustness, fairness, and sustainability. The central claim is that autonomous collaboration should be treated as an infrastructure problem. The value of a collaborative language-model system depends on whether its coordination mechanisms remain stable under stress, whether its decisions can be traced and contested, whether its resource consumption is justified, and whether its benefits and harms are distributed fairly. These concerns are not peripheral to technical performance. They are constitutive of it.

2. From Single-Agent Assistance to Multi-Agent Language Ecologies

The foundations of reinforcement learning were built around agents that learn from interaction with an environment, balancing exploration and exploitation to maximize long-term reward [1]. Deep reinforcement learning extended this paradigm to high-dimensional sensory inputs and complex control tasks, demonstrating that neural networks could approximate policies and value functions in domains such as video games and board games [2], [3]. In parallel, the transformer architecture reshaped natural language processing by enabling scalable attention-based sequence modeling [4]. Large language models then demonstrated broad few-shot capabilities, allowing a single model to perform many tasks

without task-specific training [5]. Instruction tuning and preference-based fine-tuning further aligned model behavior with human expectations [6], [7]. These developments created the conditions for language models to function as general-purpose reasoning and communication components.

Multi-agent reinforcement learning introduced a distinct set of challenges. When multiple learning agents interact, the environment becomes non-stationary because each agent's policy changes over time. Centralized training with decentralized execution, actor-critic methods, and learned communication protocols were developed to address coordination, competition, and information sharing [8], [9]. These approaches showed that agents can learn to cooperate, specialize, and communicate, but they also revealed fragility: coordination conventions can collapse, credit assignment becomes ambiguous, and equilibrium selection depends on initial conditions and reward design. Language-model agents inherit these difficulties while adding new ones. Their action spaces are semantic, their internal representations are opaque, and their outputs can be interpreted by humans in ways that influence trust and accountability [18].

Recent work on language-model agents has begun to operationalize collaboration through role assignment, conversation, memory, and reflection. Generative agents simulate believable social behavior in interactive environments [10]. ReAct combines reasoning traces with tool use, allowing agents to plan and act in external environments [16]. Reflexion uses verbal self-feedback to improve future behavior without updating model weights [17]. These techniques make it possible for language-model agents to coordinate through natural language rather than only through scalar rewards. A recent large-scale effort toward agentic models for full-scenario applications illustrates the ambition to scale such collaboration across diverse tasks and environments [11]. Yet the same flexibility that makes language-based coordination powerful also makes it difficult to specify, verify, and govern. Natural language is expressive, ambiguous, and context-dependent, so coordination failures may appear as misinterpretations, role drift, or persuasive but incorrect consensus.

3. Architecture and Coordination Structures

The architecture of a multi-agent language-model system determines how information flows, how decisions are made, and where control resides. Centralized architectures place a coordinator or orchestrator at the center. The coordinator decomposes tasks, assigns roles, aggregates results, and resolves conflicts. This structure can improve consistency and simplify oversight because a single component maintains a global view. However, it also creates a bottleneck and a single point of failure. If the coordinator is compromised, biased, or overloaded, the entire system may degrade. Centralization can also concentrate power in the hands of the organization that controls the orchestrator, raising concerns about accountability and competitive dynamics.

Decentralized architectures distribute decision-making across agents that communicate peer to peer. This can improve resilience, allow local adaptation, and reduce dependence on a central authority. In software development, for example, agents may represent product managers, programmers, testers, and reviewers who negotiate requirements and implementation details [12], [13]. Decentralization can surface diverse perspectives and reduce groupthink, but it can also produce coordination overhead, redundant work, and unresolved conflicts. Without shared conventions, agents may talk past one another or converge on norms that are efficient locally but harmful globally. The design challenge is to provide enough structure for coherence without suppressing the adaptability that makes multi-agent systems attractive.

Hierarchical architectures combine central direction with local autonomy. A high-level planner sets goals and constraints, while lower-level agents execute specialized subtasks and report outcomes. This resembles human organizations and can support scalable division of labor. MetaGPT, for instance, encodes standardized operating procedures into a multi-agent software company, showing how role specialization and structured artifacts can improve collaborative output [14]. Hierarchical systems can also support escalation, where uncertain or high-stakes decisions are referred to a more capable or more constrained agent. The trade-off is that hierarchy may reproduce bureaucratic rigidity, obscure local knowledge, and diffuse responsibility across layers. If every agent can claim to have followed instructions, no agent may be accountable for the collective result.

Market-oriented and debate-based coordination structures offer alternative mechanisms. Agents may bid for tasks, exchange resources, or compete to provide the most persuasive answer. Debate can improve reasoning by forcing agents to defend claims and expose weaknesses, but it can also reward rhetorical skill over truth. CAMEL explores communicative agents that role-play and collaborate through structured dialogue, demonstrating both the promise and the instability of open-ended language-based coordination [15]. In practice, robust systems may combine multiple structures: centralized governance for safety-critical constraints, decentralized exploration for creative tasks, and hierarchical review for quality assurance. The key insight is that architecture is not a neutral container. It shapes what agents learn, what they hide, and what humans can see.

4. Learning, Credit Assignment, and Adaptive Collaboration

Multi-agent reinforcement learning provides mechanisms for agents to improve through experience, but credit assignment remains a central difficulty. When a collaborative task succeeds or fails, which agent deserves praise or blame? Global rewards can encourage cooperation, but they may also allow free riding or obscure individual contributions. Local rewards can sharpen incentives, but they may encourage agents to optimize narrow metrics at the expense of collective goals. In language-model systems, these problems are compounded because agents produce semantic artifacts whose value is often judged by other agents or by humans with imperfect information. A reviewer agent may approve code that passes tests but contains security flaws. A planner agent may choose an efficient strategy that violates fairness constraints. The learning signal must therefore encode not only task success but also process quality, safety, and compliance with governance rules [23].

Preference-based learning and reinforcement learning from human feedback have become important tools for aligning model behavior with human values [6], [7]. Constitutional approaches extend this by using explicit principles to guide self-critique and revision [24]. These methods can be adapted to multi-agent settings, where agents critique one another's outputs against shared norms. Reflexion demonstrates that verbal feedback can serve as a lightweight learning mechanism, allowing agents to revise strategies without retraining [17]. However, verbal reinforcement is not a substitute for structural accountability. If agents generate their own critiques, they may reinforce shared biases or converge on convenient rationalizations. Human oversight must therefore be positioned not only at the end of a pipeline but also at points where objectives are set, constraints are updated, and exceptions are adjudicated.

Non-stationarity is another defining feature of multi-agent learning. As one agent improves, the best response of other agents changes. In language-model collaboration, this can lead to arms races in persuasion, role inflation, or strategic withholding of information. Agents may

learn to manipulate evaluators, exploit ambiguities in instructions, or form coalitions that exclude dissenting perspectives. These behaviors are not necessarily malicious. They can emerge from optimization pressure in complex environments. Robust design requires mechanisms that detect distributional shifts, preserve diversity of reasoning, and maintain transparency about how norms evolve. It also requires evaluation that measures collective behavior over time, including whether collaboration remains stable under personnel changes, model updates, and adversarial inputs [19].

5. Infrastructure, Deployment, and Sustainability

Autonomous language-model collaboration is infrastructure-intensive. It depends on compute clusters, inference services, data pipelines, orchestration middleware, logging systems, and human review interfaces. Each additional agent increases the number of model calls, context windows, and communication steps. This can improve coverage and reliability, but it also increases latency, cost, and energy consumption. The hidden technical debt of machine learning systems is well documented: data dependencies, configuration entanglement, and feedback loops can make systems difficult to maintain even when individual components are sound [22]. Multi-agent systems amplify this debt because interactions among agents create emergent behaviors that are not visible in any single component. A change to one agent's prompt, memory policy, or tool access can alter the behavior of the entire collective.

Sustainability must be considered at multiple levels. Environmental sustainability concerns the energy and carbon footprint of training and inference. Economic sustainability concerns the cost of running many agents for long-horizon tasks. Operational sustainability concerns the maintainability of complex orchestration logic and the availability of skilled personnel. Social sustainability concerns whether the benefits of collaboration are widely shared or concentrated among a few platform providers. Foundation models already raise questions about power concentration and environmental impact [18]. Multi-agent collaboration can intensify these pressures by multiplying inference calls and by locking organizations into proprietary orchestration layers. Resource-aware scheduling, model cascades, caching, and selective agent activation can reduce waste, but they require governance that treats efficiency as a design objective rather than an afterthought.

Deployment also raises questions about observability and reversibility. A single model call is relatively easy to log. A multi-agent interaction may involve dozens or hundreds of messages, tool calls, and state updates. Without structured traces, it becomes difficult to reconstruct why a decision was made or which agent introduced an error. Model cards and datasheets provide useful documentation practices for individual models and datasets [20], [21]. In collaborative systems, documentation must extend to interaction protocols, role definitions, escalation rules, and version histories. Deployment should include mechanisms for graceful degradation, such as fallback to human review, constrained single-agent operation, or conservative default actions when confidence is low. The infrastructure should make it easier to stop, inspect, and repair a collaborative process than to let it run unchecked because it appears productive.

6. Robustness, Security, and Failure Modes

Robustness in multi-agent language-model systems is not simply the absence of crashes. It is the ability to maintain acceptable behavior under uncertainty, adversarial pressure, and distributional shift. Concrete problems in artificial intelligence safety include reward hacking, negative side effects, scalable oversight, and safe exploration [23]. These problems reappear in collaborative settings with new intensity. Reward hacking can occur when agents optimize

a proxy metric, such as number of completed tasks, while ignoring quality or safety. Negative side effects can arise when one agent's actions degrade shared resources or violate constraints that another agent was supposed to protect. Scalable oversight becomes harder because humans cannot inspect every interaction in real time. Safe exploration becomes riskier because language-model agents may take actions in external environments through tools and application programming interfaces.

Security threats include prompt injection, data exfiltration, model inversion, and coordination attacks. In a multi-agent system, a compromised agent can influence others through persuasive messages or fabricated evidence. Agents that share memory or tools can propagate contaminated information across the collective. An adversary might not need to compromise the orchestrator; it may be enough to inject a single malicious instruction into a conversation that other agents trust. Defenses must therefore include provenance tracking, trust scoring, sandboxing, least-privilege tool access, and anomaly detection. Ethical and social risks of language models, including misinformation, discrimination, and privacy violations, are not eliminated by distributing responsibility across agents [19]. They may be harder to trace because harm emerges from interaction rather than from a single output.

Failure modes in multi-agent systems are often systemic rather than local. Agents may enter loops of mutual critique without resolution. They may converge prematurely on a consensus that reflects the most confident rather than the most accurate agent. They may escalate conflicts until a coordinator imposes a solution that ignores minority concerns. They may also fail silently, producing plausible but incorrect outputs that pass weak evaluations. Robust design requires redundancy in reasoning paths, diversity in model families or prompting strategies, and explicit mechanisms for dissent. It also requires testing under adversarial conditions, including cases where agents are incentivized to deceive, collude, or defect. Robustness is therefore a property of the socio-technical system, not a checkbox for individual models.

7. Fairness, Governance, and Policy

Fairness in autonomous collaboration involves both distributive and procedural dimensions. Distributive fairness asks who benefits from the system and who bears its costs. If multi-agent systems improve productivity in software engineering or scientific research, do those gains accrue to workers, firms, or platform owners? Procedural fairness asks whether decisions are made through transparent and contestable processes. A collaborative system that reaches a correct outcome through opaque coalition dynamics may still be unfair if affected parties cannot understand or challenge the reasoning. Bias can enter through training data, reward models, role assignments, and interaction norms. Weidinger and colleagues catalog ethical and social risks from language models, including discrimination, exclusion, and information hazards [19]. Multi-agent collaboration can amplify these risks when agents reinforce one another's biases or when a dominant agent's perspective shapes the collective output.

Governance must operate at multiple layers. At the model layer, developers can use documentation, evaluation, and alignment techniques to reduce harmful behavior [20], [24]. At the system layer, architects can define role permissions, communication rules, escalation thresholds, and audit trails. At the organizational layer, institutions can establish review boards, incident reporting, and redress mechanisms. At the policy layer, regulators can require transparency, safety testing, and accountability for high-stakes deployments. These layers interact. A strong model-level alignment process cannot compensate for a system architecture that concentrates power or hides decisions. A detailed audit trail cannot compensate for the

absence of legal accountability when harm occurs. Governance must therefore be designed as a coherent stack rather than a collection of isolated measures.

Policy implications are especially significant for cross-border and cross-sector deployments. Language-model agents may operate across jurisdictions, languages, and regulatory regimes. A collaborative system used in healthcare may be subject to privacy laws that differ from those governing financial services or education. Agents that share data across organizational boundaries may create compliance risks even if no single agent violates a rule. Policymakers may need to require documentation of agent roles, interaction protocols, and decision authority. They may also need to support standards for interoperability and auditability so that oversight does not depend on proprietary tools. The goal is not to freeze innovation but to ensure that autonomous collaboration remains legible to the publics and institutions it affects.

8. Evaluation, Observability, and Operational Governance

Evaluating multi-agent language-model collaboration requires metrics that go beyond task accuracy. Useful dimensions include coordination efficiency, communication overhead, robustness to agent failure, fairness of outcomes, interpretability of decision paths, and cost per successful task. Evaluation should also consider temporal dynamics: does the system improve with experience, or does it drift toward brittle conventions? Does it maintain diversity of perspectives, or does it collapse into groupthink? Does it handle novel situations gracefully, or does it fail catastrophically when inputs shift? These questions require longitudinal studies, stress testing, and simulation of adversarial scenarios. They also require qualitative analysis of interaction traces, because quantitative metrics may miss subtle forms of manipulation or exclusion.

Observability is the foundation of operational governance. Every agent action, message, tool call, and state change should be logged in a way that supports reconstruction and review. Logs should be tamper-evident, access-controlled, and retained according to clear policies. Human overseers need interfaces that summarize collective behavior without overwhelming them with raw transcripts. They also need the ability to intervene at multiple points: before a task begins, during coordination, and after outcomes are produced. Model cards and datasheets offer templates for documenting components and data, but collaborative systems need additional artifacts such as interaction cards, role charters, and escalation protocols [20], [21]. These artifacts make the system legible to auditors, users, and affected communities.

Operational governance also involves maintenance and change management. Multi-agent systems are not static. Models are updated, tools are added, policies change, and personnel turn over. Each change can alter emergent behavior. Organizations should therefore treat collaborative configurations as versioned systems with staged rollouts, canary deployments, and rollback plans. They should monitor for regressions in fairness, safety, and cost, not only in task success. They should also create channels for affected users to report problems and receive explanations. Hidden technical debt in machine learning systems often arises because maintenance is deferred until failure [22]. In multi-agent collaboration, deferred maintenance can produce opaque and entrenched behaviors that are difficult to unwind. Governance must therefore be continuous, not episodic.

9. Case Illustrations and Cross-Domain Comparisons

Software development has become a prominent case for multi-agent language-model collaboration. AutoGen provides a framework for multi-agent conversation and tool use [12]. ChatDev organizes agents into roles such as designer, programmer, and tester to simulate a

software company [13]. MetaGPT encodes standardized procedures and artifacts to improve coordination [14]. These systems illustrate the benefits of role specialization, iterative critique, and artifact sharing. They also illustrate risks: agents may generate plausible code that fails security tests, reviewers may approve work without deep understanding, and coordination overhead may outweigh gains for simple tasks. The software domain is attractive because success can be partially verified through tests and execution, but verification remains incomplete for maintainability, security, and user needs.

Scientific assistance offers another case. Agents can literature search, generate hypotheses, design experiments, analyze data, and draft reports. Collaboration among agents with different specialties may improve coverage and reduce individual bias. However, scientific work depends on reproducibility, peer review, and ethical oversight. If agents share unverified claims or fabricate citations, the collective output may appear rigorous while being unreliable. Cross-domain comparison with software suggests that domains with strong automated verification are more amenable to autonomous collaboration, while domains with contested evidence and high-stakes consequences require tighter human oversight. The same architectural pattern can produce different outcomes depending on the verification regime.

Public administration and interactive simulation present further contrasts. Generative agents can simulate social behavior and support participatory planning [10]. Multi-agent systems might help draft policy options, model stakeholder responses, or identify unintended consequences. Yet public decisions involve legitimacy, representation, and due process. An autonomous collaboration that excludes affected groups or hides its reasoning may undermine democratic accountability even if it produces efficient recommendations. In contrast, interactive entertainment and training simulations may tolerate more autonomy and experimentation because the consequences are contained. These comparisons show that the appropriate level of autonomy is domain-specific. It depends on reversibility, verification capacity, affected populations, and legal frameworks.

10. Conclusion

Multi-agent reinforcement learning for autonomous large language model collaboration is best understood as a system-level endeavor. The technical core involves coordination, learning, and credit assignment, but the outcomes depend on architecture, infrastructure, governance, and policy. Centralized, decentralized, hierarchical, and market-oriented structures each offer distinct advantages and vulnerabilities. Learning mechanisms must handle non-stationarity, align collective behavior with human values, and preserve accountability. Deployment must confront compute intensity, maintenance burden, and observability gaps. Robustness and security require defenses against adversarial influence and emergent failure modes. Fairness and governance demand attention to distribution, procedure, and contestability. Evaluation must measure collective behavior over time rather than only individual model accuracy.

The path forward is not to maximize the number of agents or the autonomy of their interactions. It is to design collaborative systems that remain legible, reversible, and just under real-world conditions. This requires layered governance, auditable coordination traces, resource-aware scheduling, and evaluation regimes that treat safety, fairness, and sustainability as first-class objectives. It also requires interdisciplinary collaboration among reinforcement learning researchers, systems engineers, social scientists, legal scholars, and affected communities. Autonomous language-model collaboration may become a powerful infrastructure for problem-solving, but only if its design acknowledges that collaboration is

never merely technical. It is a form of social organization, and it should be governed accordingly.

References

1. Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.
2. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529-533.
3. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... & Hassabis, D. (2017). Mastering the game of Go without human knowledge. *Nature*, 550(7676), 354-359.
4. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998-6008.
5. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877-1901.
6. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35, 27730-27744.
7. Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, 30, 4299-4307.
8. Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Abbeel, O. P., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in Neural Information Processing Systems*, 30, 6379-6390.
9. Foerster, J., Assael, I. A., de Freitas, N., & Whiteson, S. (2016). Learning to communicate with deep multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 29, 2137-2145.
10. Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*.
11. Jiang, H., Xin, G., Huang, Y., Zhu, G., Zhang, J., Lin, X., ... & Cao, Z. (2026). AgentOmnia: Scaling Agentic Models for Full-Scenario Applications. *arXiv preprint arXiv:2607.23124*.
12. Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., ... & Wang, C. (2023). AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*.
13. Qian, C., Liu, W., Liu, H., Chen, N., Dang, Y., Li, J., ... & Sun, M. (2023). ChatDev: Communicative agents for software development. *arXiv preprint arXiv:2307.07924*.

14. Hong, S., Zhuge, M., Chen, J., Zheng, X., Cheng, Y., Zhang, C., ... & Wang, Z. (2023). MetaGPT: Meta programming for a multi-agent collaborative framework. arXiv preprint arXiv:2308.00352.
15. Li, G., Hammoud, H. A. A. K., Itani, H., Khizbullin, D., & Ghanem, B. (2023). CAMEL: Communicative agents for "mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36, 51991-52008.
16. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. arXiv preprint arXiv:2210.03629.
17. Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 8634-8652.
18. Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., ... & Liang, P. (2021). On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258.
19. Weidinger, L., Mellor, J., Rauh, M., Griffin, C., Uesato, J., Huang, P. S., ... & Gabriel, I. (2021). Ethical and social risks of harm from language models. arXiv preprint arXiv:2112.04359.
20. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., ... & Gebru, T. (2019). Model cards for model reporting. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 220-229.
21. Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daume III, H., & Crawford, K. (2021). Datasheets for datasets. *Communications of the ACM*, 64(12), 86-92.
22. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., ... & Dennison, D. (2015). Hidden technical debt in machine learning systems. *Advances in Neural Information Processing Systems*, 28, 2503-2511.
23. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in AI safety. arXiv preprint arXiv:1606.06565.
24. Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., ... & Kaplan, J. (2022). Constitutional AI: Harmlessness from AI feedback. arXiv preprint arXiv:2212.08073.
25. Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., ... & Zhang, Y. (2023). Sparks of artificial general intelligence: Early experiments with GPT-4. arXiv preprint arXiv:2303.12712.