

Efficient Agent Scaling through Lightweight Large Language Model Adaptation

Bina Apte

Department of Computer Science and Engineering, University of Nevada, Reno, Reno, NV, USA.

binaa@unr.edu

Napoleone Maspero

School of Information Technology, University of Cincinnati, Cincinnati, OH, USA.

napoleonemail@uc.edu

Sigfried Hofmann

Department of Computer Science, University of New Hampshire, Durham, NH, USA.

sigfried.hofmann@unh.edu

Abstract

The rapid emergence of language-model-based agents has shifted artificial intelligence from isolated prediction tasks toward persistent, tool-using, and socially embedded computational actors. Scaling such agents efficiently is not simply a matter of increasing parameter counts or deploying more inference replicas; it requires a systems-level rethinking of adaptation, orchestration, memory, governance, and infrastructure. This paper examines lightweight large language model adaptation as a strategic pathway for agent scaling. We argue that low-rank adaptation, prompt and prefix tuning, retrieval augmentation, modular skill libraries, and compact policy heads can reduce the marginal cost of creating specialized agents while preserving enough general competence for open-ended interaction. The analysis connects these adaptation choices to structural trade-offs in latency, energy consumption, robustness, fairness, and institutional accountability. We further consider multi-agent architectures, where lightweight customization enables heterogeneous populations of agents rather than a single monolithic model. Such populations introduce new failure modes, including error propagation, coordination collapse, opaque responsibility chains, and unequal access to adaptation capacity. Through cross-domain examples in software engineering, scientific discovery, customer operations, and public-sector services, the paper shows that efficient agent scaling depends less on one optimization technique than on an integrated socio-technical architecture. We conclude with a research agenda for sustainable, auditable, and equitable agent ecosystems.

Keywords

agent scaling, lightweight adaptation, large language models, multi-agent systems, AI governance, sustainable infrastructure, fairness, robustness.

1. Introduction

Large language models have become general-purpose substrates for reasoning, generation, and interaction, a trajectory anticipated by foundational work on transformer architectures [1], bidirectional pretraining [2], few-shot learning [3], and foundation models [4]. As these models are embedded in software agents, the unit of analysis shifts from a single model output to an ongoing process of perception, planning, tool use, memory revision, and social

coordination. This shift creates a scaling problem that is qualitatively different from conventional model scaling. Increasing the size of a central model may improve average capability, but it does not automatically produce efficient specialization, reliable multi-agent coordination, or affordable long-horizon operation. The costs of inference, latency, energy, and operational risk grow with every additional agent instance. Consequently, the question of how to scale agents efficiently becomes a question about architecture and governance as much as about model design.

Lightweight adaptation offers one response to this problem. Rather than training a separate large model for every domain, organization, or task, lightweight methods adjust a shared base model through small parameter updates, soft prompts, retrieval indices, adapters, or external skill modules. This approach resonates with earlier critiques of unchecked model growth, including concerns about environmental and social costs [5], technical debt in machine learning systems [6], and safety problems that arise when optimization objectives are underspecified [7]. It also intersects with ethical risk analyses that emphasize downstream harms, opacity, and accountability gaps [8]. If agent scaling is pursued only through larger central models, those risks may be concentrated and amplified. If it is pursued through lightweight adaptation, the risks become distributed across many adapted endpoints, each of which may be cheaper to create but harder to monitor.

The central argument of this paper is that efficient agent scaling requires a layered socio-technical architecture. At the model layer, lightweight adaptation reduces the cost of specialization. At the orchestration layer, routing, memory, and tool interfaces determine whether specialization translates into reliable behavior. At the infrastructure layer, serving systems, energy markets, and hardware availability shape what can be sustained. At the governance layer, audit trails, model documentation, and policy controls determine whether scaled agent populations remain legible to institutions and affected communities. The paper develops this argument by examining theoretical foundations, architectural patterns, deployment concerns, robustness and fairness, governance implications, and cross-domain illustrations. It treats efficiency not as a narrow computational metric but as a composite property involving economic viability, environmental sustainability, institutional trust, and equitable access.

2. Theoretical and Systems Foundations of Lightweight Adaptation

The conceptual basis for lightweight adaptation lies in the observation that general-purpose pretraining produces representations that can be reused across many downstream tasks. Early transfer learning and contextual representation work demonstrated that a single pretrained model could support diverse applications with relatively modest task-specific modification [2]. Subsequent scaling studies clarified that model performance improves predictably with data, compute, and parameters, but also that optimal allocation depends on the training regime and deployment context [23]. The compute-optimal perspective further showed that many models were undertrained relative to their size, implying that data and adaptation strategy can matter as much as raw parameter count [24]. These findings support a shift from monolithic scaling toward portfolio-based scaling, where a shared base model is paired with many lightweight specializations.

Lightweight adaptation can take several forms. Parameter-efficient fine-tuning adjusts a small subset of weights or inserts trainable modules while freezing most of the base model. Prompt and prefix tuning condition the model through learned continuous prompts without changing its core weights. Retrieval augmentation externalizes knowledge into searchable stores,

allowing an agent to access current or domain-specific information without retraining. Tool use and skill libraries allow agents to call external functions, databases, or simulators, as illustrated by work on reasoning and acting [13] and self-taught tool use [14]. Self-consistency and reflection methods improve reliability by sampling multiple reasoning paths or using verbal feedback to revise behavior [15], [16]. Each method shifts cost away from full retraining, but each also introduces new dependencies, failure modes, and governance requirements.

From a systems perspective, the important trade-off is not whether lightweight adaptation is cheaper in isolation, but how it changes the total cost of ownership. A small adapter may reduce training cost, yet it can increase inference complexity if the serving system must load many adapters, route requests, and maintain versioned retrieval indices. A retrieval-based agent may reduce fine-tuning needs, yet it can increase latency and expose sensitive data through search logs. A tool-using agent may avoid encoding procedural knowledge in weights, yet it can fail catastrophically when external APIs change or return adversarial content. These considerations connect to hidden technical debt in machine learning systems, where entanglement, undeclared consumers, and configuration drift accumulate silently [6]. Efficient agent scaling therefore requires lifecycle accounting rather than a narrow focus on training expense.

The theoretical foundation also includes representational limits. A lightweight adapter cannot create capabilities that the base model lacks entirely, and it may amplify biases already present in pretraining data. Fairness research has long shown that unaware optimization can reproduce or intensify discrimination when protected attributes correlate with proxy variables [9]. In agent settings, this risk becomes procedural: an adapted agent may choose which tools to use, which users to prioritize, and which information to surface. Model documentation practices such as model cards can improve transparency, but they must be extended to cover adaptation provenance, retrieval sources, and multi-agent interaction histories [25]. Without such extensions, lightweight adaptation may produce a proliferation of small, undocumented behavioral policies that are difficult to audit collectively.

3. Architectural Patterns for Scalable Agent Populations

The architecture of an agent population determines whether lightweight adaptation produces efficiency or fragmentation. A centralized pattern uses one large model with many prompts or adapters and a shared orchestration layer. This pattern simplifies version control and central auditing but concentrates inference load and creates a single point of failure. A federated pattern distributes adapted models across edge devices, departments, or regions. It improves privacy and latency for local tasks but complicates consistency, security, and governance. A hierarchical pattern combines a generalist coordinator with specialist agents, allowing the coordinator to decompose tasks and delegate to lightweight domain agents. This pattern resembles multi-agent frameworks such as AutoGen and CAMEL, which structure conversations among role-specific agents [20], [21]. It can improve modularity, but it also introduces error propagation and coordination overhead.

A key architectural decision is how agents share memory. Stateless agents are easier to replicate and audit, but they lose continuity and cannot learn from long-horizon interaction. Stateful agents can maintain episodic memory, user preferences, and task histories, as demonstrated by generative agent simulations [18]. However, shared memory creates privacy risks, stale information, and poisoning vulnerabilities. A lightweight adaptation strategy might therefore separate private memory, team memory, and public knowledge stores, each with

different access controls and retention policies. The design must also specify how memory is summarized, when it is invalidated, and who can inspect or delete it. These are governance questions as much as engineering questions, because memory shapes the effective behavior of an agent even when its model weights remain unchanged.

Another architectural pattern is the society of minds approach, where multiple language-model agents debate, critique, and refine one another’s outputs [19]. Such systems can improve reasoning and reduce individual bias, but they can also produce conformity, amplifying the most confident rather than the most accurate voice. Lightweight adaptation can support this pattern by giving each agent a distinct role, expertise, or perspective through small parameter updates or role-specific prompts. Yet the resulting population may become difficult to evaluate as a whole. A system with dozens of adapted agents may pass individual tests while failing collectively due to coordination collapse, resource contention, or inconsistent policy interpretations. Efficient scaling therefore requires population-level evaluation, not only per-agent benchmarking.

Recent work on full-scenario agentic models illustrates the ambition and the difficulty of scaling agents across many application domains [17]. Such efforts suggest that broad competence can be pursued through unified training and orchestration, but they also raise questions about whether a single large agentic model should dominate all scenarios. Lightweight adaptation offers a complementary route: a strong general base model provides common sense and language ability, while small adapters, retrieval indices, and tools provide domain specificity. The optimal balance depends on task volatility, privacy constraints, latency budgets, and regulatory exposure. In high-stakes domains, a more centralized and heavily audited architecture may be justified. In low-stakes, high-volume domains, a federated population of lightweight agents may be more sustainable. The architectural choice is therefore contextual rather than universal.

4. Infrastructure, Deployment, and Operational Sustainability

Infrastructure is the material substrate of agent scaling. Even lightweight adaptation depends on compute for training, storage for adapters and retrieval indices, network bandwidth for tool calls, and energy for inference. The environmental critique of large models remains relevant because aggregate demand can grow even when per-request cost falls [5]. If lightweight adaptation lowers the barrier to creating new agents, it may increase the total number of agents and thus total energy consumption. This rebound effect is a central concern for sustainable deployment. Efficient scaling must therefore be measured at the fleet level, not only at the level of a single model invocation. Metrics such as energy per task, carbon intensity by region, and hardware utilization over time provide a more realistic picture.

Serving systems for adapted agents face distinctive challenges. A production platform may host thousands of adapters, each with different compatibility requirements, security policies, and update schedules. Dynamic loading and unloading of adapters can reduce memory pressure but introduce latency spikes and cache misses. Retrieval indices must be updated frequently, and stale indices can produce confident but incorrect answers. Tool calls introduce external dependencies that may be rate-limited, versioned, or unavailable. These dynamics resemble the technical debt patterns described in machine learning systems, where glue code, pipeline jungles, and configuration debt accumulate [6]. A sustainable agent infrastructure must treat adaptation artifacts as first-class assets with versioning, testing, rollback, and deprecation procedures.

Deployment also raises questions of access and equity. Lightweight adaptation can democratize agent creation because it reduces the cost of specialization. Small organizations, community groups, and individual developers can build agents without training a foundation model from scratch. However, they still depend on base models, cloud services, and tool ecosystems controlled by a small number of providers. If adaptation capacity is cheap but base model access is expensive or restricted, then the benefits of lightweight scaling may concentrate among those who already control infrastructure. Fairness through awareness suggests that system designers should examine how benefits and burdens are distributed, not merely whether the underlying algorithm is neutral [9]. In agent ecosystems, this means asking who can adapt, who can audit, who bears the cost of errors, and who has recourse when harm occurs.

Operational sustainability also involves human labor. Agents do not eliminate work; they redistribute it toward data curation, prompt engineering, adapter maintenance, evaluation design, incident response, and user support. As agent populations grow, the human oversight burden can grow with them. A lightweight adaptation strategy that creates many specialized agents may require a correspondingly distributed team of domain experts and auditors. If those experts are not available, the system may default to automated evaluation that misses context-specific harms. The infrastructure plan must therefore include staffing, training, and escalation pathways. Efficiency claims that ignore human maintenance costs are incomplete and can lead to brittle deployments.

5. Robustness, Safety, and Fairness in Distributed Agent Ecosystems

Robustness in agent systems is not only about model accuracy. It concerns the ability to detect and recover from adversarial inputs, tool failures, memory corruption, and coordination errors. A lightweight adapter may be robust in one domain but fragile when the task distribution shifts. A retrieval-augmented agent may be vulnerable to poisoned documents. A tool-using agent may be induced to call unsafe functions or leak sensitive data. Reflection and self-consistency can improve reliability, but they can also create loops, increase cost, and produce false confidence [15], [16]. Safety research has long emphasized specification gaming, reward hacking, and negative side effects [7]. In multi-agent settings, these problems become relational: one agent’s action changes the environment for others, and failures can cascade.

Ethical risk frameworks for language models identify harms such as misinformation, discrimination, privacy violation, and psychological manipulation [8]. Agent scaling multiplies the contexts in which these harms can occur. A single adapted customer service agent may be harmless in isolation, but a population of agents interacting with vulnerable users, competing for attention, and making consequential recommendations can produce systemic effects. Fairness auditing becomes more complex because harms may emerge from interactions among agents rather than from any single model. For example, a scheduling agent may consistently assign less desirable shifts to a demographic group because of proxies in historical data, while a separate approval agent reinforces the pattern through opaque scoring. Addressing such harms requires interaction-level auditing and clear responsibility assignment.

Transparency is a necessary but insufficient response. Model cards provide structured documentation of intended use, limitations, and performance across groups [25]. For adapted agents, documentation should also include adaptation data sources, tool permissions, memory retention rules, evaluation coverage, and known failure modes. Yet documentation can become a compliance exercise that does not change behavior. Robust governance must

connect documentation to enforcement: access controls, rate limits, audit logs, and rollback capabilities. It must also provide meaningful notice and contestability for affected users. In public-sector deployments, for instance, an agent that denies benefits or prioritizes cases must offer an explanation and a human appeal path. Without such mechanisms, efficiency gains may come at the cost of legitimacy.

Fairness in agent scaling also concerns representation in adaptation data. Lightweight adaptation often relies on smaller, domain-specific datasets, which may not reflect the diversity of the user population. A model adapted on data from one region, language, or demographic group may perform poorly for others. This is not a reason to abandon lightweight adaptation, but it is a reason to require disaggregated evaluation and inclusive data collection. The fairness through awareness literature emphasizes that decision-makers should consider protected attributes to avoid disparate impact when base rates differ [9]. In agent systems, this principle can be translated into requirements for context-aware evaluation, community participation, and impact assessment before deployment. Efficiency and fairness are not inherently opposed, but they require deliberate architectural choices.

6. Governance, Policy, and Institutional Implications

Governance for scaled agent populations must operate across multiple levels. At the organizational level, policies should define which agents can access which data, which tools they may invoke, and which decisions require human approval. At the platform level, providers should offer audit interfaces, usage logs, and safety filters that are consistent across adapters. At the regulatory level, rules may need to address transparency, liability, data protection, and sector-specific risks. The challenge is that lightweight adaptation blurs the boundary between model provider and application developer. If an organization fine-tunes a base model with a small adapter, who is responsible for a harmful output? The base model provider, the adapter creator, the orchestrator, or the deployer? Clear allocation of responsibility is essential, but current legal and policy frameworks often lag behind technical practice.

Policy design must also account for the global distribution of compute and data. Efficient adaptation could enable more localized and culturally specific agents, which may improve representation. However, it could also enable rapid creation of deceptive or manipulative agents at low cost. Governance therefore needs both enabling and protective measures. Enabling measures include open standards for adapters, interoperable evaluation tools, and public compute resources for under-resourced communities. Protective measures include know-your-customer requirements for high-risk agents, watermarking or provenance tracking, and restrictions on automated decision-making in domains such as criminal justice, health, and immigration. The goal is not to slow innovation uniformly but to shape the direction of scaling toward socially beneficial uses.

Institutional capacity is a limiting factor. Regulators, auditors, and civil society organizations may not have the technical expertise or computational resources to evaluate complex agent populations. A single audit of a large model is already difficult; auditing thousands of lightweight adaptations is far harder. This asymmetry favors large providers unless governance infrastructure is designed to be accessible. Third-party auditing, standardized reporting, and independent red-teaming can help, but they require funding and legal protection. Public agencies may also need their own technical staff to assess procurement proposals and monitor deployed systems. Without such capacity, policy will rely on self-

reporting and voluntary compliance, which is unlikely to be sufficient for high-stakes applications.

The sustainability of agent scaling is therefore inseparable from institutional design. If efficiency gains are captured by a few firms while risks are externalized to users and communities, the overall system becomes less resilient. A more balanced approach would treat adaptation infrastructure as a shared resource with public-interest obligations. This could include open benchmarks for robustness and fairness, shared repositories of safe adapters, and requirements for incident reporting. It could also include participatory design processes that involve affected communities in defining acceptable agent behavior. Such measures may appear to slow deployment, but they can reduce the likelihood of costly failures and erosion of trust. Efficient scaling should mean scaling what works, not merely scaling what is technically possible.

7. Case Illustrations and Cross-Domain Comparisons

Software engineering provides an instructive case. Agents can assist with code generation, testing, debugging, and documentation, as suggested by work on evaluating large language models trained on code [22]. A lightweight adaptation strategy might create specialized agents for different programming languages, frameworks, or codebases. These agents can be integrated into development environments and retrieve project-specific context. However, they can also introduce insecure code, propagate outdated patterns, or create licensing issues. The efficiency of adaptation must be weighed against the cost of reviewing and maintaining generated code. In this domain, robust tool integration and version control are as important as model quality.

Scientific discovery offers another case. Agents can search literature, propose hypotheses, run simulations, and analyze data. Lightweight adaptation allows a base model to specialize in chemistry, biology, climate science, or materials research without full retraining. Retrieval augmentation connects agents to current publications and databases. Tool use allows them to call specialized simulators. Yet scientific agents must be calibrated for uncertainty and reproducibility. A confident but incorrect hypothesis can waste research resources or spread misinformation. Governance in this domain requires provenance tracking, peer review, and clear separation between exploratory generation and validated findings. The efficiency of agent scaling depends on whether the surrounding scientific infrastructure can absorb and verify a higher volume of candidate ideas.

Customer operations and public services illustrate the social dimension. In customer operations, lightweight agents can handle routine inquiries, route complex cases, and personalize interactions. The efficiency gains can be substantial, but failures affect real people. A billing agent that misinterprets a policy or a support agent that cannot escalate a complaint can cause financial and emotional harm. In public services, agents may determine eligibility, schedule appointments, or provide information about legal rights. Fairness, transparency, and appeal mechanisms are essential. The same lightweight adaptation that makes localization affordable can also make inconsistent behavior harder to detect across many offices or jurisdictions. Cross-domain comparison shows that the appropriate level of centralization varies with stakes, scale, and regulatory context.

These cases share a common lesson: lightweight adaptation is most effective when paired with strong orchestration and governance. A domain-specific adapter alone does not produce a reliable agent. It must be connected to accurate retrieval, safe tools, clear escalation paths,

and continuous evaluation. The surrounding architecture determines whether efficiency gains translate into better outcomes or simply faster failure. Cross-domain comparison also reveals that the most successful deployments are likely to be those that treat agents as part of a broader workflow rather than as autonomous replacements for human judgment. This is consistent with the systems view that technical components and social processes co-determine performance.

8. Future Directions and Research Agenda

Several research directions follow from this analysis. First, evaluation must move from single-model benchmarks to population-level metrics that capture coordination, resource use, fairness across groups, and long-horizon reliability. Second, adaptation methods should be designed with governance in mind, producing artifacts that are versioned, auditable, and easy to revoke. Third, infrastructure research should develop efficient serving systems for large numbers of adapters and retrieval indices, with attention to energy and latency. Fourth, policy research should clarify liability and accountability across the adaptation supply chain. Fifth, participatory methods should involve affected communities in defining acceptable agent behavior and redress mechanisms.

A particularly important direction is the development of shared adaptation commons. Just as open-source software libraries accelerated development, a curated commons of safe, well-documented adapters and tools could reduce duplication and improve quality. Such a commons would require licensing, security review, and maintenance. It could also create new forms of dependency and lock-in. The design must balance openness with accountability. Public funding and academic participation could help ensure that the commons serves diverse needs rather than only commercial priorities. This is a socio-technical design problem, not merely a repository problem.

Another direction concerns the interaction between lightweight adaptation and continual learning. Agents that operate over long periods will encounter new information, user feedback, and environmental changes. They may need to update their adapters, retrieval stores, or policies without catastrophic forgetting or uncontrolled drift. Research on safe continual learning, memory management, and change detection is therefore central. The goal is not to make every agent learn continuously in an unconstrained way, but to design controlled update pathways with human oversight. This connects to broader questions about autonomy, consent, and the right to explanation. Efficient scaling should not come at the expense of the ability to understand and correct agent behavior.

9. Conclusion

Efficient agent scaling through lightweight large language model adaptation is a promising but demanding socio-technical strategy. It can reduce the cost of specialization, enable heterogeneous agent populations, and support localized and domain-specific applications. However, it also distributes risk, complicates governance, and may increase aggregate resource consumption if not carefully managed. The paper has argued that efficiency must be understood at the fleet level, encompassing training, inference, memory, tools, human oversight, and environmental impact. Architectural choices about centralization, memory, and coordination shape whether lightweight adaptation produces resilience or fragmentation. Robustness, fairness, and accountability cannot be added after deployment; they must be designed into the adaptation lifecycle.

The path forward requires integrated research across machine learning, systems engineering, policy, and social science. Technical advances in parameter-efficient tuning, retrieval, and tool use must be matched by advances in evaluation, auditing, and institutional capacity. Governance should enable beneficial innovation while protecting against concentrated power and unaccounted harm. If agent scaling is pursued solely as a race toward larger models and more autonomous deployments, the benefits may be uneven and the risks systemic. If it is pursued as a deliberate effort to build sustainable, auditable, and equitable agent ecosystems, lightweight adaptation can contribute to a more resilient digital infrastructure. The central challenge is not whether agents can be scaled, but how they should be scaled and for whom.

References

1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998–6008.
2. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT 2019*, 4171–4186.
3. 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.
4. 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*.
5. Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 610–623.
6. 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.
7. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.
8. 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*.
9. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012). Fairness through awareness. *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, 214–226.
10. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. *Proceedings of the 38th International Conference on Machine Learning*, 8748–8763.
11. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., ... & Lample, G. (2023). LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

12. 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.
13. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. *International Conference on Learning Representations*.
14. Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., ... & Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.
15. Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., ... & Zhou, D. (2023). Self-consistency improves chain of thought reasoning in language models. *International Conference on Learning Representations*.
16. 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.
17. 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*.
18. 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. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 1–22.
19. Zhuge, M., Wang, W., Kirsch, L., Faccio, F., Khizbullin, D., & Schmidhuber, J. (2023). Mindstorms in natural language-based societies of mind. *arXiv preprint arXiv:2305.17066*.
20. 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.
21. 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 framework. *arXiv preprint arXiv:2308.08155*.
22. Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. O., Kaplan, J., ... & Zaremba, W. (2021). Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
23. Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... & Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.
24. Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., ... & Sifre, L. (2022). Training compute-optimal large language models. *Advances in Neural Information Processing Systems*, 35, 30016–30030.
25. 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.