

Adaptive Reasoning Path Selection for Resource-Constrained Large Language Models

Kashish Sarna

Department of Electrical Engineering and Computer Science, University of Missouri,
Columbia, MO, USA.

kashishmail@missouri.edu

Abstract

Large language models have become general-purpose computational artifacts, yet their most capable reasoning behaviors are often entangled with substantial memory, latency, energy, and governance costs. Resource-constrained settings, including edge devices, regulated enterprise enclaves, classrooms, clinics, public-sector terminals, and intermittently connected field environments, cannot assume unlimited accelerator capacity or persistent cloud connectivity. This paper examines adaptive reasoning path selection as a system-level strategy for reconciling reasoning quality with constrained computation. Rather than treating reasoning as a monolithic capability, the paper conceptualizes it as a portfolio of selectable trajectories whose depth, branching, tool use, retrieval intensity, and verification overhead can be adjusted at runtime. The analysis connects model architecture, inference serving, memory hierarchy, adaptation methods, and orchestration policy to the practical question of which reasoning path should be chosen under a given resource envelope. It also considers structural trade-offs among accuracy, robustness, fairness, latency, maintainability, and sustainability. The paper argues that adaptive path selection cannot be reduced to a model-side optimization because deployment context, infrastructure ownership, monitoring, fallback behavior, and accountability determine whether such systems remain trustworthy. Governance implications are examined through transparency, contestability, procurement, and environmental accounting. The conclusion is that resource-constrained reasoning systems require adaptive, auditable, and institutionally embedded selection mechanisms, not merely smaller models.

Keywords

adaptive reasoning, large language models, resource-constrained computing, system architecture, governance, sustainability, robustness, fairness.

1. Introduction

Large language models have moved from research prototypes to infrastructural components of search, education, software development, administration, and scientific assistance [1], [2]. Their capacity to perform few-shot inference, synthesize heterogeneous information, and generate plausible action plans has encouraged institutions to embed them in workflows that previously depended on human judgment or specialized software [3], [4]. However, the most visible demonstrations of reasoning capability often rely on generous computational budgets, high-bandwidth connectivity, and centralized serving infrastructure. When these assumptions fail, the resulting systems may become brittle, expensive, or exclusionary. A model that performs well in a data center may be unusable on a low-power laptop, an off-grid clinic, or a municipal office with strict procurement and privacy constraints. The central problem is therefore not only whether a model can reason, but whether it can select an appropriate form of reasoning under real constraints.

Adaptive reasoning path selection addresses this problem by treating reasoning as a dynamically configurable process rather than a fixed property of the model. A system may choose among direct answer generation, short intermediate computation, multi-step deliberation, self-consistency sampling, tool invocation, retrieval augmentation, or external verification. These paths differ not only in expected accuracy but also in memory pressure, latency, energy consumption, failure modes, and explainability. The selection mechanism may operate at the level of a request, a user session, a device class, or an institutional policy. In some cases, a lightweight path is sufficient; in others, a more expensive path is justified because the cost of error is high. The design challenge is to make that choice without creating opaque or unfair tiering among users.

This paper develops a system-level account of adaptive reasoning path selection for resource-constrained large language models. It emphasizes architecture, infrastructure, deployment, sustainability, robustness, fairness, and policy rather than narrow model benchmarking. The argument is that adaptive selection is best understood as a socio-technical control problem involving model capabilities, serving stacks, hardware constraints, human oversight, and institutional governance. The paper proceeds by reviewing related work, defining resource-constrained reasoning as a systems problem, outlining an architectural perspective, analyzing trade-offs, discussing robustness and fairness, examining deployment and sustainability, and considering governance and policy. It concludes that adaptive path selection can improve access and efficiency, but only if it is accompanied by transparency, fallback mechanisms, and accountability structures.

2. Background and Related Work

The transformer architecture established the foundation for contemporary large language models by enabling scalable attention-based sequence processing [1]. Subsequent work demonstrated that increasing model scale and training data could produce strong few-shot behavior across tasks [2], while broader analyses of foundation models highlighted both their generality and their systemic risks [3], [4]. Reasoning-specific methods then sought to improve performance by eliciting intermediate steps, sampling multiple reasoning traces, or organizing deliberation as a search process [5], [6], [7]. These methods often improve accuracy but also multiply computation, memory use, and latency. A recent line of work on trajectory-level policy optimization and alignment for small-parameter models illustrates how reasoning path filtering can be integrated into training and inference pipelines [8]. That direction is relevant because it treats path selection as a learnable or policy-guided decision rather than an ad hoc prompt choice.

Augmented language models extend reasoning beyond the parameters of the model by invoking retrieval systems, calculators, code interpreters, and external tools [9], [10]. Embodied and multimodal systems further show that reasoning can be distributed across perception, planning, and action modules [11]. Open-weight models have made local experimentation more feasible [12], while parameter-efficient adaptation and quantization methods reduce the cost of specialization and deployment [13], [14], [15]. Serving research has also advanced memory management, batching, and offloading strategies for generative inference under hardware constraints [16], [17], [18]. Together, these developments suggest that resource-constrained reasoning is not a single technical bottleneck but a composition of model design, runtime orchestration, and infrastructure provisioning.

At the same time, critical scholarship has warned that scale alone does not resolve social and environmental concerns. Work on stochastic parrots, risk taxonomies, and algorithmic

monoculture has shown that large models can reproduce harms, concentrate power, and homogenize outcomes [19], [20], [21]. Ethical principles and governance guidelines have proliferated, yet principles alone often fail to translate into enforceable practice [22], [23]. Energy and carbon accounting for training and inference further complicates claims of efficiency [24], [25]. Adaptive reasoning path selection must therefore be evaluated not only by whether it saves computation, but by whether it distributes saved computation fairly, preserves contestability, and avoids shifting burdens to users or communities with less institutional power.

3. Resource-Constrained Reasoning as a Systems Problem

Resource constraints are often described in terms of memory, compute, or battery life, but in practice they are relational and contextual. A hospital may have sufficient hardware yet prohibit sending patient data to external servers. A rural school may have intermittent connectivity and shared devices. A public agency may face procurement rules that favor auditable local systems over opaque cloud services. These constraints shape which reasoning paths are permissible, not merely which are technically possible. Adaptive path selection must therefore begin with an inventory of constraints: hardware capacity, network availability, privacy requirements, latency tolerances, cost ceilings, maintenance expertise, and legal obligations. A path that is optimal in a benchmark environment may be unacceptable in a regulated or low-resource setting.

The systems view also reveals that reasoning quality is not a scalar property. A direct answer may be fast but uninspectable; a chain-of-thought path may be more accurate but also more verbose and potentially misleading; a tool-augmented path may improve factual grounding but introduce dependency on external services; a verification path may reduce error but increase latency and cost. The selection mechanism must decide which dimension matters most for the request at hand. For a casual question, speed and fluency may dominate. For a medical dosage calculation or a legal deadline, verification and auditability may dominate. For a classroom exercise, pedagogical transparency may dominate. Adaptive selection thus requires a policy that maps context to path, not a universal preference for maximum reasoning depth.

A further systems issue is state management. Reasoning paths are not isolated operations; they create intermediate artifacts such as retrieved documents, tool outputs, partial chains, cached embeddings, and versioned prompts. In constrained environments, these artifacts compete for memory and storage. If the system retains too much, it may exhaust device resources or create privacy risks. If it retains too little, it may fail to support audit, debugging, or continuity across sessions. The architecture must therefore define what is stored locally, what is summarized, what is discarded, and what is transmitted. These decisions are not merely engineering details; they determine whether users can contest an outcome or understand why a particular answer was produced.

4. Architecture for Adaptive Path Selection

An adaptive reasoning architecture can be understood as a layered control system. The lowest layer contains the model or models available for inference, possibly in multiple sizes and quantization levels. The middle layer contains orchestration components that classify requests, estimate resource budgets, select reasoning paths, invoke tools, and manage caches. The upper layer contains policy, monitoring, and governance functions that define allowable behavior, escalation rules, and accountability. This layering matters because path selection is

not solely a model decision. It is a negotiated outcome among technical capability, operational policy, and user context. A well-designed system makes these negotiations visible enough to be audited.

At the orchestration layer, request classification can be lightweight and approximate. The system may use heuristics such as input length, domain, presence of numerical content, ambiguity, or known risk categories. It may also use a small classifier or a distilled model to estimate whether a request requires deeper reasoning. The goal is not perfect prediction but bounded decision-making under uncertainty. If the classifier is uncertain, the system can choose a conservative path, ask a clarifying question, or default to a verified but slower route. In resource-constrained settings, the cost of misclassification must be considered in both directions: under-reasoning may produce harmful errors, while over-reasoning may exhaust battery or bandwidth and delay critical tasks.

Path execution itself can be modular. A direct generation path may use a small quantized model. A retrieval path may query a local index before generation. A tool path may call a calculator, database, or rule engine. A deliberation path may generate multiple candidate answers and compare them. A verification path may invoke a separate checker or human review. The architecture should allow these modules to be composed and substituted without redesigning the entire system. This modularity supports graceful degradation: if a network connection fails, the system can fall back to local retrieval or direct generation; if memory pressure rises, it can reduce sampling; if a tool is unavailable, it can flag uncertainty rather than fabricate an answer.

5. Trade-Offs in Computation, Memory, and Latency

The most immediate trade-off in adaptive reasoning is between accuracy and resource consumption. Deeper reasoning often improves performance on complex tasks, but it increases the number of tokens generated, the number of model invocations, and the amount of intermediate state retained. Sampling multiple trajectories further multiplies cost [6], while tree-like search expands the space of possible paths [7]. In a data center, these costs may be absorbed through batching and parallelism [17], [18]. On a mobile device or edge server, they may exceed memory or thermal limits. Adaptive selection is therefore a form of budgeting: the system allocates a finite reasoning budget across requests, users, and time windows.

Latency introduces a second trade-off. Users may prefer a fast approximate answer, but some tasks require slower verification. In interactive settings, the system can stream partial results, present a provisional answer, and refine it if the user requests more depth. This approach can preserve responsiveness while still allowing deeper reasoning when needed. However, it also creates risks: users may accept the provisional answer without waiting for verification, or they may not understand that the answer is incomplete. The interface must therefore communicate uncertainty and reasoning status without overwhelming the user. In constrained environments, latency is not only a technical metric; it affects whether a tool is usable in time-sensitive workflows such as triage, emergency response, or public service delivery.

Memory and storage trade-offs are equally important. Quantization and parameter-efficient adaptation reduce model footprint [13], [14], [15], but they may alter reasoning behavior in ways that are uneven across tasks. Caching can improve latency but consumes memory and may retain sensitive content. Offloading can extend capability but depends on network reliability and may introduce privacy risks [16]. The architecture must decide which assets are persistent, which are session-scoped, and which are ephemeral. These decisions should be

documented because they affect reproducibility, audit, and user trust. A resource-constrained system that silently discards evidence or compresses context may be efficient, but it may also be unable to justify its outputs.

6. Robustness, Fairness, and Evaluation

Robustness in adaptive reasoning means maintaining acceptable behavior across distribution shifts, hardware failures, network interruptions, and adversarial inputs. A path selection policy trained in one environment may fail in another. For example, a policy that assumes reliable cloud access may collapse when connectivity drops. A policy that relies on a particular tool may propagate that tool's errors. Robustness therefore requires fallback paths, uncertainty estimation, and monitoring for degraded performance. It also requires testing under realistic constraints rather than only on curated benchmarks. Evaluation should include edge cases, long-tail domains, multilingual inputs, and low-resource user contexts. Otherwise, the system may appear effective while systematically failing those who depend on constrained infrastructure.

Fairness is closely related. If deeper reasoning paths are reserved for users with faster devices, better connectivity, or premium accounts, the system may create a two-tier epistemic environment. Users with fewer resources may receive less verified, less transparent, or less accurate answers. This is not only a market issue; it is a governance issue because access to reasoning quality can affect education, health, employment, and civic participation. Adaptive selection should therefore include equity constraints. For example, certain high-stakes requests may require a minimum verification path regardless of device class. Alternatively, the system may queue expensive reasoning for later execution when resources become available, rather than denying it outright. The design should make such trade-offs explicit and subject to review.

Evaluation must also move beyond aggregate accuracy. Metrics should include calibration, abstention behavior, explanation quality, latency percentiles, energy per successful task, and disparity across user groups. A model that is slightly less accurate overall but much more consistent across contexts may be preferable in public infrastructure. Similarly, a path that is slower but auditable may be appropriate for regulated decisions. Evaluation should be participatory where possible, involving affected communities and domain experts. In resource-constrained settings, the people closest to the constraints often have the most useful knowledge about acceptable trade-offs. Treating them as test subjects rather than co-designers risks producing systems that are technically efficient but socially misaligned.

7. Deployment, Infrastructure, and Sustainability

Deployment determines whether adaptive reasoning remains viable over time. A system that works in a pilot may fail when scaled to many users, updated models, or new regulations. Infrastructure choices such as on-device inference, edge servers, private cloud, or public cloud shape latency, privacy, cost, and control. On-device inference offers autonomy and data locality but is limited by hardware. Edge servers can balance locality and capacity but require physical maintenance and security. Cloud services offer elasticity but may create dependency and recurring costs. A hybrid architecture can combine these options, but it also increases operational complexity. The selection policy must account for these infrastructure realities, not just model-level performance.

Sustainability is a central but often overlooked dimension. Training large models consumes significant energy and emits carbon, and inference at scale can rival or exceed training costs

depending on usage [24], [25]. Adaptive path selection can reduce inference energy by avoiding unnecessary deliberation, but it can also encourage more frequent use by making systems seem cheaper. This rebound effect means that efficiency gains may not translate into absolute reductions. Institutions should therefore monitor total consumption, not just per-request cost. They should also consider embodied emissions from hardware, e-waste from rapid replacement cycles, and the water and land impacts of data centers. Sustainability is not a side constraint; it is part of the resource envelope that defines what responsible reasoning looks like.

Maintainability is another deployment concern. Adaptive systems accumulate policies, prompts, tools, classifiers, caches, and fallback rules. Without documentation and version control, these components can drift and produce inconsistent behavior. A path that was safe last year may become unsafe after a model update or a change in external data. Maintenance requires observability, logging, rollback capability, and clear ownership. In under-resourced organizations, maintenance capacity may be the scarcest resource of all. A system that requires constant expert tuning may not be sustainable, even if it performs well initially. Simplicity, modularity, and transparent defaults are therefore not merely aesthetic preferences; they are conditions for long-term reliability.

8. Governance and Policy Implications

Governance begins with transparency about when and why a reasoning path is selected. Users may not need to see every internal step, but they should know whether an answer was verified, retrieved, generated directly, or produced with tool support. They should also know whether a cheaper or faster path was chosen because of resource limits. Without this information, they cannot calibrate trust or decide when to seek human review. Transparency should be accompanied by contestability: a mechanism for users to request a deeper path, challenge an outcome, or obtain an explanation. In high-stakes domains, contestability may need to be a legal or contractual requirement rather than a voluntary feature.

Procurement and policy also shape adaptive reasoning. Public institutions may require data localization, audit trails, accessibility, and nondiscrimination. They may prohibit certain forms of automated decision-making or require human oversight. These requirements should be translated into technical constraints within the path selection policy. For example, a policy may specify that requests involving benefits eligibility cannot use a direct generation path without human confirmation. Another policy may require that reasoning traces be retained for a defined period. The challenge is to make these rules enforceable and testable. Principles alone cannot guarantee ethical AI [22], and the global landscape of ethics guidelines shows broad agreement on values but weaker agreement on implementation [23]. Adaptive systems need concrete governance hooks.

Accountability is complicated by distributed responsibility. A path may involve a model provider, a device manufacturer, a tool vendor, a data source, and a deploying institution. If an error occurs, who is responsible? The governance framework should assign duties for monitoring, disclosure, correction, and redress. It should also address power asymmetries. Resource-constrained users may have little ability to negotiate terms or switch providers. Policy interventions such as interoperability requirements, open standards, public auditing, and support for local capacity can reduce dependency. Without such measures, adaptive reasoning may reproduce existing inequalities under the guise of efficiency.

9. Case Illustrations and Cross-Domain Comparisons

Consider a rural clinic using a local language model for preliminary triage support. The clinic may have a modest edge server, intermittent connectivity, and strict patient privacy rules. A direct generation path may be acceptable for general health information, but dosage or emergency advice may require retrieval from a curated local formulary and a verification path. If the edge server is overloaded, the system may defer deeper reasoning or refer to a human clinician. The adaptive policy must prioritize safety over speed in high-risk cases while avoiding unnecessary delays for routine questions. It must also log enough information to support clinical audit without storing more patient data than necessary.

In education, a school may use adaptive reasoning to support tutoring. Here, the goal is not only correct answers but also pedagogical value. A direct answer may be efficient but unhelpful if it bypasses learning. A stepwise path may be more appropriate, but it may also expose students to misleading or overly complex reasoning. The selection mechanism might choose a path based on the student's prior performance, the subject, and the available device. Fairness concerns arise if students with older devices receive less interactive support. The governance challenge is to define a minimum pedagogical standard that applies regardless of resource constraints.

In public administration, a municipal office may use a model to summarize regulations or draft responses. The consequences of error can be legal and reputational. A retrieval-heavy path with citations may be preferred over free-form generation. If the office lacks cloud connectivity, the system may need a local index and a smaller model. The path selection policy might require that any answer citing law be traceable to a specific retrieved passage. If no passage is found, the system should abstain rather than speculate. This case illustrates how resource constraints and accountability can align: a more constrained path may actually be more auditable and legally defensible.

10. Future Directions

Future work should develop selection policies that are both adaptive and accountable. This includes methods for uncertainty-aware routing, energy-aware scheduling, and cross-device orchestration. It also includes evaluation frameworks that measure fairness and robustness under realistic constraints. Researchers should study how users understand different reasoning paths and how interface design affects trust. More work is needed on local-first architectures, privacy-preserving retrieval, and federated or decentralized adaptation. These directions are not purely technical; they require collaboration with policy experts, domain practitioners, and affected communities.

Another direction is the standardization of reasoning metadata. If systems recorded which path was used, what resources were consumed, what sources were consulted, and what uncertainty remained, then auditors and users could compare outcomes across contexts. Such metadata could support procurement, incident review, and sustainability reporting. It could also enable interoperability between different model providers and deployment environments. The challenge is to design metadata that is useful without being excessive or privacy-invasive. A minimal, well-defined schema may be more effective than comprehensive logging.

Finally, research should examine the political economy of adaptive reasoning. Efficiency improvements may consolidate power if they favor large providers with sophisticated orchestration infrastructure. Conversely, adaptive path selection could support decentralization if it enables smaller models, local tools, and community-controlled infrastructure to participate. The outcome depends on policy choices, standards, and

investment. The technical community cannot assume that efficiency automatically produces equity. Adaptive reasoning must be designed with explicit attention to who benefits, who bears risk, and who can contest decisions.

11. Conclusion

Adaptive reasoning path selection offers a practical and conceptually rich response to the constraints facing large language models in real-world deployment. By treating reasoning as a portfolio of selectable trajectories, systems can balance accuracy, latency, memory, energy, and auditability according to context. This paper has argued that such selection is fundamentally a systems problem. It requires layered architecture, resource-aware orchestration, robust fallback behavior, fair evaluation, sustainable infrastructure, and governance mechanisms that make decisions visible and contestable. It also requires humility about prediction: no classifier can perfectly know when deeper reasoning is necessary, so systems must be designed to fail gracefully and involve humans when stakes are high.

The promise of adaptive selection is that it can make capable reasoning more accessible to resource-constrained environments. The risk is that it can create opaque tiers of service, shift costs to users, or optimize for narrow efficiency while ignoring fairness and environmental impact. The difference depends on design and policy. Institutions should not treat path selection as a purely internal optimization. They should define minimum standards, document trade-offs, monitor disparities, and provide routes for appeal. Researchers should develop methods that are not only accurate but also accountable, sustainable, and inclusive. If these conditions are met, adaptive reasoning path selection can support a more resilient and equitable relationship between large language models and the varied infrastructures on which people depend.

References

1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998-6008.
2. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877-1901.
3. Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunsell, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X. L., Li, X., Ma, T., Malik, A., Manning, C. D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J. C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J. S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich,

- R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Shoker, S., Siken, S., Sohl-Dickstein, J., Stein, A., Suhr, A., Suzuki, A., Taori, R., Thomas, A. W., Tramer, F., Wang, R. E., Wang, W., Wu, B., Xiong, C., Xie, S. M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhao, Y., Zheng, K., Zhou, K., & Liang, P. (2021). On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258.
4. Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J. Y., & Wen, J. R. (2023). A survey of large language models. arXiv preprint arXiv:2303.18223.
 5. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35, 24824-24837.
 6. Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., Chowdhery, A., & Zhou, D. (2023). Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*.
 7. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., & Narasimhan, K. (2023). Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.
 8. Li, Q. (2026, May). TrajLite PPO: Scalable Reasoning Path Filtering and Alignment for Small Parameter Large Language Models. In *2026 2nd International Conference on Artificial Intelligence and Digital Ethics (ICAIDE)* (pp. 29-32). IEEE.
 9. Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.
 10. Mialon, G., Dessì, R., Lomeli, M., Nalmpantis, C., Pasunuru, R., Raileanu, R., Celikyilmaz, A., Grave, E., & Scialom, T. (2023). Augmented language models: A survey. arXiv preprint arXiv:2302.07842.
 11. Driess, D., Xia, F., Sajjadi, M. S. M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., Huang, W., Chebotar, Y., Sermanet, P., Duckworth, D., Levine, S., Vanhoucke, V., Hausman, K., Toussaint, M., Greff, K., Zeng, A., Mordatch, I., & Florence, P. (2023). PaLM-E: An embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning* (pp. 8469-8488).
 12. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023). LLaMA: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.
 13. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.
 14. Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs. *Advances in Neural Information Processing Systems*, 36.

15. Frantar, E., Ashkboos, S., Hoefler, T., & Alistarh, D. (2023). GPTQ: Accurate post-training quantization for generative pre-trained transformers. In International Conference on Learning Representations.
16. Sheng, Y., Zheng, L., Yuan, B., Li, Z., Ryabinin, M., Chen, B., Liang, P., Ré, C., Stoica, I., & Zhang, C. (2023). High-throughput generative inference of large language models with a single GPU. In Proceedings of the 40th International Conference on Machine Learning (pp. 31094-31116).
17. Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., & Stoica, I. (2023). Efficient memory management for large language model serving with PagedAttention. In Proceedings of the 29th Symposium on Operating Systems Principles (pp. 611-626).
18. Pope, R., Douglas, S., Chowdhery, A., Devlin, J., Bradbury, J., Heek, J., Xiao, K., Agrawal, S., & Dean, J. (2023). Efficiently scaling transformer inference. Proceedings of Machine Learning and Systems, 5, 606-624.
19. Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (pp. 610-623).
20. Weidinger, L., Uesato, J., Rauh, M., Griffin, C., Huang, P. S., Mellor, J., Glaese, A., Cheng, M., Balle, B., Kasirzadeh, A., Biles, C., Brown, S., Kenton, Z., Hawkins, W., Stepleton, T., Birhane, A., Hendricks, L. A., Rimell, L., Isaac, W., Haas, J., Legassick, S., Irving, G., & Gabriel, I. (2022). Taxonomy of risks posed by language models. In Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (pp. 214-229).
21. Bommasani, R., Creel, K. A., Kumar, A., Hudson, D. A., & Liang, P. (2022). Picking on the same person: Does algorithmic monoculture lead to outcome homogenization? Advances in Neural Information Processing Systems, 35, 3663-3678.
22. Mittelstadt, B. (2019). Principles alone cannot guarantee ethical AI. Nature Machine Intelligence, 1(11), 501-507.
23. Jobin, A., Ienca, M., & Vayena, E. (2019). The global landscape of AI ethics guidelines. Nature Machine Intelligence, 1(9), 389-399.
24. Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 3645-3650).
25. Luccioni, A. S., Viguier, S., & Ligozat, A. L. (2023). Estimating the carbon footprint of BLOOM, a 176B parameter language model. Journal of Machine Learning Research, 24(253), 1-15.