

Adaptive Inference Optimization for Large Language Models Using Reinforcement Learning

Kamila Koscielak

Department of Computer Science, University of Alabama at Birmingham, Birmingham, AL, USA.

contactkamila@uab.edu

Fem Koning

Department of Computer Science, University of North Texas, Denton, TX, USA.

femkoning@unt.edu

Baljiwan L. Gandhi

Department of Computer Science, George Mason University, Fairfax, VA, USA.

blgandhi@gmu.edu

Abstract

Large language models have moved from research prototypes to densely interconnected production infrastructures, where inference cost, latency, reliability, and governance now determine whether advanced capabilities can be delivered equitably and sustainably. Static serving configurations, fixed batching policies, and hand-tuned deployment heuristics struggle to accommodate heterogeneous workloads, shifting hardware conditions, and evolving user expectations. This paper examines adaptive inference optimization for large language models through reinforcement learning, treating the serving stack as a socio-technical control system rather than a narrow model-compression problem. The discussion integrates scaling behavior, speculative decoding, memory management, quantization, tool use, and reasoning-time compute allocation with institutional concerns such as auditability, fairness, robustness, and environmental sustainability. Reinforcement learning is positioned as a mechanism for learning policies that dynamically allocate compute, select decoding strategies, manage cache and batching behavior, and route requests across heterogeneous accelerators. Yet adaptive control also introduces new failure modes, including reward misspecification, feedback loops, opaque optimization, and unequal service quality. The paper argues that technically effective inference optimization must be paired with governance mechanisms, observability standards, and policy frameworks that make trade-offs explicit. It concludes that the most credible path forward is not maximal automation but accountable adaptivity, where learned policies remain interpretable, bounded, and aligned with public-interest constraints.

Keywords

adaptive inference; large language models; reinforcement learning; model serving; infrastructure governance; fairness; sustainability.

1. Introduction

Large language models have become a central computational substrate for contemporary artificial intelligence, enabling conversational agents, code synthesis, document analysis, scientific assistance, and multimodal interaction. The transformer architecture established the

foundational mechanism for parallel sequence modeling and attention-based representation learning [1], while large-scale pretraining demonstrated that broad linguistic and world knowledge could emerge from next-token prediction over vast corpora [2]. Subsequent alignment techniques, particularly reinforcement learning from human feedback, transformed raw generative capacity into more instruction-following and usable systems [3]. These developments have been accompanied by a shift from isolated model artifacts toward foundation-model ecosystems that support many downstream applications, organizational dependencies, and public-facing services [4]. As these systems diffuse, the central engineering challenge is no longer only whether a model can produce a plausible response, but whether it can do so at acceptable latency, cost, reliability, and social cost.

Inference has therefore become the operational bottleneck and the main economic surface of large language model deployment. Training a frontier model is expensive, but serving millions or billions of requests can dominate long-term expenditure because each interaction requires computation across many parameters, memory movement, and orchestration across accelerators. The serving stack must simultaneously manage prompt processing, token generation, batching, caching, scheduling, autoscaling, and failure recovery. Static configurations are poorly matched to this environment because request lengths, task difficulty, user priority, hardware availability, thermal conditions, and network latency vary continuously. A policy that maximizes throughput during a batch-processing window may be inappropriate for an interactive assistant, while a policy tuned for minimal latency may waste energy and capacity. Adaptive inference optimization responds to this mismatch by allowing the system to adjust its behavior in real time.

Reinforcement learning is a natural candidate for this control problem because it addresses sequential decision-making under uncertainty. Instead of relying solely on fixed rules, a learned policy can observe system state, choose actions, receive feedback, and improve over time. In the context of large language model inference, actions may include selecting a decoding strategy, choosing a quantization path, deciding whether to invoke speculative execution, allocating additional reasoning steps, routing a request to a particular accelerator pool, or adjusting batch size and cache retention. The reward signal may combine latency, throughput, accuracy, energy consumption, fairness constraints, and service-level objectives. However, this framing immediately raises structural questions: who defines the reward, how are trade-offs audited, what happens when optimization degrades minority workloads, and how can adaptive systems remain stable under adversarial or anomalous conditions. This paper argues that adaptive inference optimization must be understood as a governance problem embedded in infrastructure, not merely as an algorithmic improvement.

2. Foundations of Large Language Model Inference as a Socio-Technical System

The computational profile of large language model inference is shaped by scaling laws and architectural choices. Empirical scaling research showed that model performance improves predictably with increases in parameters, data, and compute, which encouraged the construction of larger and more capable systems [5]. Later work on compute-optimal training clarified that model size and data volume must be balanced to avoid inefficient allocation of training resources [6]. These findings indirectly influence inference because larger models may require more memory bandwidth, more accelerators, and more sophisticated parallelism. Efficiently scaling transformer inference therefore becomes a systems problem involving partitioning, kernel optimization, memory hierarchy management, and request scheduling [7].

The inference stack is not a neutral pipeline; it encodes assumptions about workload homogeneity, acceptable latency, and the economic value of each request.

In production environments, inference delay is not a single quantity but a composite of queueing delay, prefill computation, decoding time, memory transfer, network round trips, and post-processing. Predicting these delays accurately is difficult because workloads are bursty and hardware performance can vary due to contention, thermal throttling, or background maintenance. Machine learning approaches to inference delay prediction illustrate how operational telemetry can be used to anticipate bottlenecks and inform scheduling decisions [8]. Such prediction is valuable because adaptive control depends on foresight: a policy must estimate the consequences of routing, batching, or compression choices before committing to them. Nevertheless, prediction models are themselves fallible and may encode historical biases, especially when certain request classes are underrepresented in telemetry. A socio-technical perspective requires that delay prediction be treated as an accountable infrastructure component rather than an invisible optimization detail.

The social dimension of inference infrastructure appears in service-level expectations and resource distribution. A public-facing chatbot, a hospital decision-support tool, a legal research platform, and an internal code assistant may all rely on the same underlying model family, yet their latency tolerances, accuracy requirements, privacy constraints, and failure consequences differ sharply. A uniform serving policy cannot satisfy all of these contexts without either overprovisioning or sacrificing quality. Adaptive inference offers a way to differentiate service, but differentiation can become discriminatory if premium users receive faster and more accurate responses while others receive degraded outputs. The design of the inference control layer therefore intersects with questions of fairness, transparency, and accountability long before policy intervention occurs. The technical architecture distributes attention, compute, and latency, and those distributions have social meaning.

3. Reinforcement Learning as an Adaptive Control Layer for Inference

Reinforcement learning provides a formal framework for learning sequential decisions through interaction with an environment [9]. In deep reinforcement learning, neural networks approximate policies or value functions, enabling control in high-dimensional settings such as games and simulated robotics [10]. These methods have already influenced large language model development through reinforcement learning from human feedback, where human preferences are used to shape model behavior beyond supervised fine-tuning [11]. The same conceptual machinery can be repurposed for inference optimization, but the environment differs from traditional reinforcement learning benchmarks. The state space includes queue lengths, memory utilization, request characteristics, accelerator health, energy prices, and service-level commitments. The action space includes scheduling, routing, compression, speculative decoding, and compute allocation. The reward is a negotiated objective rather than a natural score.

A reinforcement learning controller for inference could learn to balance competing objectives by observing telemetry and adjusting policies. For example, during periods of low load, it might favor higher-quality decoding, larger models, or additional reasoning steps. During peak load, it might route simple requests to smaller or quantized models, enable aggressive batching, or reduce speculative branching. The appeal of reinforcement learning lies in its ability to discover non-obvious policies that outperform static heuristics. However, the same flexibility creates risk. If the reward emphasizes average latency, the policy may neglect tail latency, which is often more important for user trust. If the reward emphasizes throughput, the

system may sacrifice accuracy for users whose requests are difficult or unusual. If the reward includes energy consumption without safeguards, the policy may shift cost to externalized environmental harm or defer computation in ways that violate service expectations.

The governance of learned inference policies therefore requires more than reward engineering. It requires constraint specification, policy evaluation, rollback mechanisms, and audit trails. A reinforcement learning agent should not be allowed to redefine the meaning of acceptable service in pursuit of a proxy objective. Instead, it should operate within bounded action spaces and explicit service guarantees. For instance, a policy might be permitted to choose among approved decoding strategies but forbidden from silently downgrading requests that carry safety-critical labels. It might be allowed to adjust batching within a latency budget but required to preserve prioritization for emergency or accessibility-related traffic. These constraints transform reinforcement learning from an opaque optimizer into a governed component of the serving infrastructure. The central design question is not whether reinforcement learning can improve inference efficiency, but under what institutional conditions its improvements remain legitimate.

4. Architectural Trade-Offs in Adaptive Inference

Speculative decoding illustrates the kind of architectural trade-off that adaptive policies must manage. A smaller draft model can propose tokens that a larger target model verifies, potentially reducing the number of expensive forward passes required per generated token [12]. Similar speculative sampling methods accelerate decoding by balancing proposal quality against verification cost [13]. These techniques are not universally beneficial. When the draft model aligns well with the target model and the workload is predictable, speculative decoding can significantly improve latency. When alignment is poor or verification overhead is high, it can waste computation and increase tail latency. An adaptive controller can decide when speculation is worth enabling based on request type, current load, and observed acceptance rates. This decision requires continuous measurement and cannot be reduced to a permanent configuration.

Memory management is another critical dimension. Large language model serving is often constrained by key-value cache growth, which limits batch size and concurrency. Efficient memory management systems such as PagedAttention treat cache blocks as manageable units, enabling more flexible sharing and allocation across requests [14]. Attention kernel optimization, exemplified by FlashAttention, reduces memory traffic and improves hardware utilization by restructuring computation around input and output awareness [15]. These innovations create opportunities for adaptive inference because they change the cost profile of different scheduling decisions. A policy that understands memory pressure can choose when to share cache, when to evict, when to batch, and when to route a request to a less loaded replica. However, aggressive cache sharing may raise privacy concerns if one request influences another through residual state, and aggressive eviction may harm users with long conversational contexts. Architectural optimization thus creates governance questions about isolation, retention, and acceptable degradation.

The trade-offs extend to model selection and routing. In a heterogeneous deployment, requests may be served by different model sizes, adapters, or quantization levels. A small model may be sufficient for classification, extraction, or simple summarization, while complex reasoning may require a larger model or additional reasoning steps. Adaptive routing can reduce cost and latency by matching request difficulty to model capacity. Yet difficulty estimation is uncertain, and errors may disproportionately affect users who phrase requests in

nonstandard ways or operate in low-resource languages. A reinforcement learning policy trained on aggregate reward may learn to route those users to cheaper models if their complaints are not represented in the feedback signal. Architectural efficiency and equitable service quality are therefore intertwined. The serving stack must expose enough information for policy evaluation while protecting privacy and preventing manipulation.

5. Quantization, Tool Use, and Reasoning-Time Allocation

Quantization is a central technique for reducing inference cost by representing weights or activations with lower precision. Work on eight-bit matrix multiplication demonstrated that large transformer inference can be made more practical without severe quality loss [16]. Post-training quantization methods such as GPTQ and SmoothQuant further expanded the design space by enabling accurate compression and activation smoothing for large models [17][18]. These techniques are attractive for adaptive inference because they offer multiple operating points: a system can choose a more aggressive quantization path under load and a more conservative path when quality matters most. The challenge is that quality degradation is not uniform across tasks. A quantized model may perform well on general conversation but poorly on numerical reasoning, legal citation, or low-resource translation. An adaptive controller must therefore estimate task sensitivity rather than treating quality as a single scalar.

Tool use and reasoning-time allocation introduce additional adaptive dimensions. Models can learn to call external tools, retrieve information, or execute code, extending their capabilities beyond parametric memory [19]. Agentic frameworks such as ReAct interleave reasoning and action, allowing models to gather information and revise plans [20]. Chain-of-thought prompting elicits step-by-step reasoning and can improve performance on complex tasks [21]. From an inference perspective, these methods allow the system to spend more compute only when needed. A simple factual query might require a single forward pass, while a multi-step planning problem might benefit from tool calls, retrieval, and extended reasoning. Reinforcement learning can learn policies that decide when to invoke these expensive behaviors. However, more reasoning is not always better. It can increase latency, expose sensitive intermediate content, amplify errors, and create new attack surfaces. Adaptive inference must therefore distinguish between useful deliberation and computational waste.

The integration of quantization, tool use, and reasoning-time allocation creates a combinatorial optimization problem. Each request could be served by many possible pipelines, and each pipeline has different latency, cost, accuracy, privacy, and energy implications. Static rules cannot capture this complexity. Reinforcement learning can, in principle, learn context-dependent policies that outperform hand-tuned heuristics. But the reward signal must be carefully designed to avoid perverse incentives. If the system rewards short responses, it may truncate necessary explanations. If it rewards tool calls, it may overuse external services. If it rewards confidence, it may encourage overconfident answers. A robust adaptive inference system needs calibrated uncertainty, human oversight for high-stakes decisions, and mechanisms for users to request higher-quality processing. The control layer should be an enabler of appropriate service, not a hidden arbiter of what users deserve.

6. Deployment, Infrastructure, and Operational Governance

Adaptive inference is inseparable from deployment operations. Machine learning systems accumulate hidden technical debt through entanglement, undeclared consumers, configuration drift, data dependencies, and feedback loops [22]. A reinforcement learning controller can worsen this debt if it is introduced without observability, versioning, and reproducible

evaluation. Operators need to know why a policy chose a particular action, what data informed that choice, and how the policy behaves when telemetry is missing or corrupted. Model cards and datasheets provide precedents for documenting model capabilities, limitations, and training data [23][25]. Similar documentation practices are needed for adaptive inference policies: their objectives, constraints, action spaces, failure modes, and evaluation results should be recorded and auditable. Without such documentation, organizations may deploy powerful optimizers that no one fully understands or can contest.

Safety and robustness are also operational concerns. Concrete problems in artificial intelligence safety include reward hacking, negative side effects, scalable oversight, and safe exploration [24]. These problems appear in inference optimization when a policy discovers shortcuts that satisfy metrics while harming users. For example, a controller might reduce average latency by dropping long-context requests, degrading responses for users who need detailed analysis. It might reduce energy consumption by caching stale results, producing outdated or contextually inappropriate answers. It might improve throughput by grouping requests in ways that leak information across tenants. These failure modes are not hypothetical; they follow from the structure of proxy objectives. A governed adaptive system must include anomaly detection, human review, red-teaming, and the ability to disable learned policies and revert to conservative fallbacks.

Infrastructure governance also involves cross-organizational dependencies. Many organizations rely on external model providers, cloud accelerators, and managed inference services. Adaptive optimization may be controlled by the provider, the customer, or a shared responsibility model. This fragmentation complicates accountability. If a provider's reinforcement learning scheduler degrades service for a particular customer segment, the customer may lack visibility into the cause. If a customer applies its own routing policy on top of a provider's API, the combined behavior may be difficult to predict. Governance frameworks should therefore specify transparency requirements for inference service-level objectives, disclosure of adaptive behavior, and mechanisms for appeal when automated optimization produces harmful outcomes. Policy interventions may include audit rights, incident reporting, and minimum service guarantees for essential applications.

7. Robustness, Fairness, and Sustainability

Fairness in adaptive inference is not limited to model outputs. It also concerns access to compute, latency, and quality. If reinforcement learning optimizes for revenue or engagement, it may learn to allocate premium resources to users who are most likely to convert or complain, while quietly degrading service for others. Such behavior can be especially harmful in education, healthcare, public benefits, and civic information, where equal access to reliable language assistance matters. Fairness-aware reinforcement learning must therefore incorporate constraints that prevent unacceptable disparities in latency, accuracy, and refusal rates across demographic, linguistic, and geographic groups. These constraints cannot be evaluated only in aggregate; they require disaggregated telemetry and careful attention to intersectional harms. Privacy-preserving measurement techniques may be necessary to detect disparities without exposing sensitive user data.

Robustness under adversarial conditions is equally important. An attacker might manipulate request patterns to steer a reinforcement learning scheduler into inefficient or unfair behavior. They might poison telemetry, flood the system with expensive requests, or exploit speculative decoding to amplify resource consumption. A robust controller should be resilient to distribution shift, resistant to reward hacking, and capable of recognizing when its

assumptions no longer hold. It should also maintain graceful degradation: if the adaptive layer fails, the system should fall back to safe, predictable behavior rather than collapsing. Robustness is not merely a technical property; it is a commitment to users that the service will remain dependable even when optimization encounters unexpected conditions.

Sustainability adds another dimension. Large language model inference consumes significant energy and contributes to data center demand, water use, and carbon emissions. Adaptive optimization can reduce waste by avoiding unnecessary computation, selecting efficient hardware paths, and scheduling flexible workloads when energy is cleaner. However, efficiency gains can be offset by rebound effects: if inference becomes cheaper, demand may increase, leading to higher total consumption. Reinforcement learning policies that optimize only for operational cost may ignore embodied carbon, hardware replacement cycles, and regional energy inequality. Sustainable inference governance should therefore consider total lifecycle impacts, not just per-request latency. It should also avoid shifting environmental burden to communities that receive little benefit from the services being optimized. Sustainability is not a secondary metric; it is a constraint on legitimate scaling.

8. Cross-Domain Comparisons and Forward-Looking Perspectives

Comparisons with other adaptive infrastructures clarify the stakes. Cloud computing uses autoscaling, load balancing, and service-level agreements to manage heterogeneous demand. Content delivery networks use edge caching and request routing to reduce latency. Financial systems use risk controls and circuit breakers to prevent cascading failures. Each domain combines automation with governance, because unconstrained optimization can produce instability or inequity. Large language model inference should learn from these precedents. It needs autoscaling that respects fairness, routing that preserves privacy, and circuit breakers that prevent runaway reasoning or speculative execution. Reinforcement learning can provide adaptive intelligence, but it should operate within a broader control architecture that includes deterministic safeguards, human oversight, and transparent reporting.

A forward-looking research agenda should pursue several directions. First, reward modeling for inference should incorporate multi-objective constraints, uncertainty, and distributional fairness rather than collapsing everything into a single score. Second, evaluation should move beyond benchmark accuracy and average latency to include tail behavior, cross-group disparities, energy use, and failure recovery. Third, infrastructure should expose standardized telemetry and policy interfaces so that adaptive behavior can be audited across organizational boundaries. Fourth, governance should define accountability for automated optimization, including disclosure, redress, and limits on high-stakes downgrading. Fifth, research should explore human-in-the-loop adaptation, where operators and affected communities can shape the objectives that inference policies pursue. These directions connect technical design with policy legitimacy.

The future of adaptive inference may involve hierarchical control. Low-level policies could manage batching, cache eviction, and kernel selection at millisecond scales. Mid-level policies could route requests and select model variants at second scales. High-level policies could negotiate service-level objectives, energy budgets, and fairness constraints at organizational scales. Reinforcement learning could operate at each level, but the levels must be aligned through shared constraints and transparent objectives. A purely bottom-up optimizer may exploit loopholes, while a purely top-down rulebook may be too rigid for dynamic conditions. Hierarchical governance offers a way to combine local adaptability with

system-wide accountability. It also creates a need for interfaces that translate technical metrics into policy-relevant language and vice versa.

9. Conclusion

Adaptive inference optimization for large language models using reinforcement learning promises substantial improvements in latency, throughput, cost, and energy efficiency. By learning policies that respond to workload conditions, hardware variability, and task complexity, serving systems can move beyond static configurations and hand-tuned heuristics. Speculative decoding, memory-efficient attention, quantization, tool use, and reasoning-time allocation provide a rich action space for such policies. Yet the same flexibility that makes reinforcement learning attractive also introduces risks of reward misspecification, opacity, unfair service differentiation, and infrastructure fragility. The most important challenge is therefore not simply to build more capable optimizers, but to embed them in accountable socio-technical systems. Adaptive inference should be bounded by explicit service guarantees, audited through disaggregated evaluation, documented like other high-impact models, and governed through clear responsibility structures.

This paper has argued that adaptive inference is an infrastructural and political matter as much as a machine learning problem. The distribution of compute, latency, and quality across users reflects institutional priorities, whether or not those priorities are stated. Reinforcement learning can help systems respond to complexity, but it cannot decide what counts as fair, safe, or sustainable without human deliberation. The path forward is accountable adaptivity: learned policies that improve efficiency while remaining interpretable, constrained, reversible, and aligned with public-interest values. If the field pursues optimization without governance, it risks creating powerful but opaque systems that quietly reshape access to language technologies. If it pursues governance without optimization, it risks leaving efficiency and sustainability gains unrealized. The credible alternative is to design inference infrastructures where reinforcement learning serves clearly articulated social and operational objectives, and where those objectives remain open to contestation.

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. 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., ... Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.
3. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35, 27730–27744.
4. Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon,

- R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., ... Liang, P. (2021). On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258.
5. Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). Scaling laws for neural language models. arXiv preprint arXiv:2001.08361.
 6. Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., ... Sifre, L. (2022). Training compute-optimal large language models. arXiv preprint arXiv:2203.15556.
 7. 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.
 8. Zhang, J., & Gao, H. (2026, March). LLM inference delay prediction based on machine learning algorithms. In *2026 International Conference on Generative Artificial Intelligence and Information Security (GAIIS)* (pp. 37-41). IEEE.
 9. Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press.
 10. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533.
 11. 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.
 12. Leviathan, Y., Kalman, M., & Matias, Y. (2023). Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning* (Vol. 202, pp. 19274–19286). PMLR.
 13. Chen, C., Borgeaud, S., Irving, G., Lespiau, J.-B., Sifre, L., & Jumper, J. (2023). Accelerating large language model decoding with speculative sampling. arXiv preprint arXiv:2302.01318.
 14. 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). ACM.
 15. Dao, T., Fu, D. Y., Ermon, S., Rudra, A., & Ré, C. (2022). FlashAttention: Fast and memory-efficient exact attention with IO-awareness. *Advances in Neural Information Processing Systems*, 35, 16344–16359.
 16. Dettmers, T., Lewis, M., Belkada, Y., & Zettlemoyer, L. (2022). LLM.int8(): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35, 30318–30332.

17. Frantar, E., Ashkboos, S., Hoefler, T., & Alistarh, D. (2023). GPTQ: Accurate post-training quantization for generative pre-trained transformers. arXiv preprint arXiv:2210.17323.
18. Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., & Han, S. (2023). SmoothQuant: Accurate and efficient post-training quantization for large language models. In Proceedings of the 40th International Conference on Machine Learning (Vol. 202, pp. 38087–38099). PMLR.
19. 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. arXiv preprint arXiv:2302.04761.
20. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2022). ReAct: Synergizing reasoning and acting in language models. arXiv preprint arXiv:2210.03629.
21. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems, 35, 24824–24837.
22. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. Advances in Neural Information Processing Systems, 28, 2503–2511.
23. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). Model cards for model reporting. In Proceedings of the Conference on Fairness, Accountability, and Transparency (pp. 220–229). ACM.
24. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in AI safety. arXiv preprint arXiv:1606.06565.
25. Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé III, H., & Crawford, K. (2021). Datasheets for datasets. Communications of the ACM, 64(12), 86–92.