

SUSTAINABLE DATA CENTERS ROADMAP

CHAPTER 2.2

Software

*Alp Kucukelbir
and Minjue Wu*

October 2025



2.2 Software

Alp Kucukelbir and Minjue Wu

A. How Is AI Software Different? _____	3
B. What Does Efficiency Mean for AI Systems? _____	4
C. AI Model Training: Is Larger Always Better? _____	4
D. AI Inference: How “Reasoning” Changed Dynamics _____	6
E. Reducing Emissions Through Flexible AI Computation _____	7
F. An Outlook on AI Software Efficiency _____	9
G. Barriers and Risks _____	11
H. Recommendations _____	12
I. References _____	14

Algorithms instruct computers on how to solve problems. But not all algorithms are created equal. Computer scientists continuously seek cleverer algorithms to improve speed and efficiency.

Consider the task of sorting a list of random numbers from lowest to highest. One approach compares every pair of adjacent numbers and swaps them if they are in the wrong order. For a list of one thousand numbers, the processor would have to make one million comparisons. There is, however, a better way: repeatedly divide the list in half and merge the smaller lists back together in sorted order. This algorithm requires only about ten thousand comparisons—a 100x speedup.

Both algorithms produce the correct ordering of numbers, from lowest to highest. But one requires less computation and energy to do so; it is an objectively better algorithm.

Clever algorithm design, as in the example above, has enabled technological innovations such as digital audio and video, medical imaging, telecommunications, DNA sequencing, and supply chain optimization. When and how such algorithmic breakthroughs arise are hard to predict—these moments are considered pivotal in computer science.

This chapter examines how software efficiency impacts data center energy consumption, from algorithmic optimizations that reduce computational requirements

to operational techniques that align artificial intelligence (AI) workloads with clean energy availability. As AI inference demand increases with autonomous agents and reasoning models, mastering both technical and operational efficiency becomes essential for sustainable data center operations.

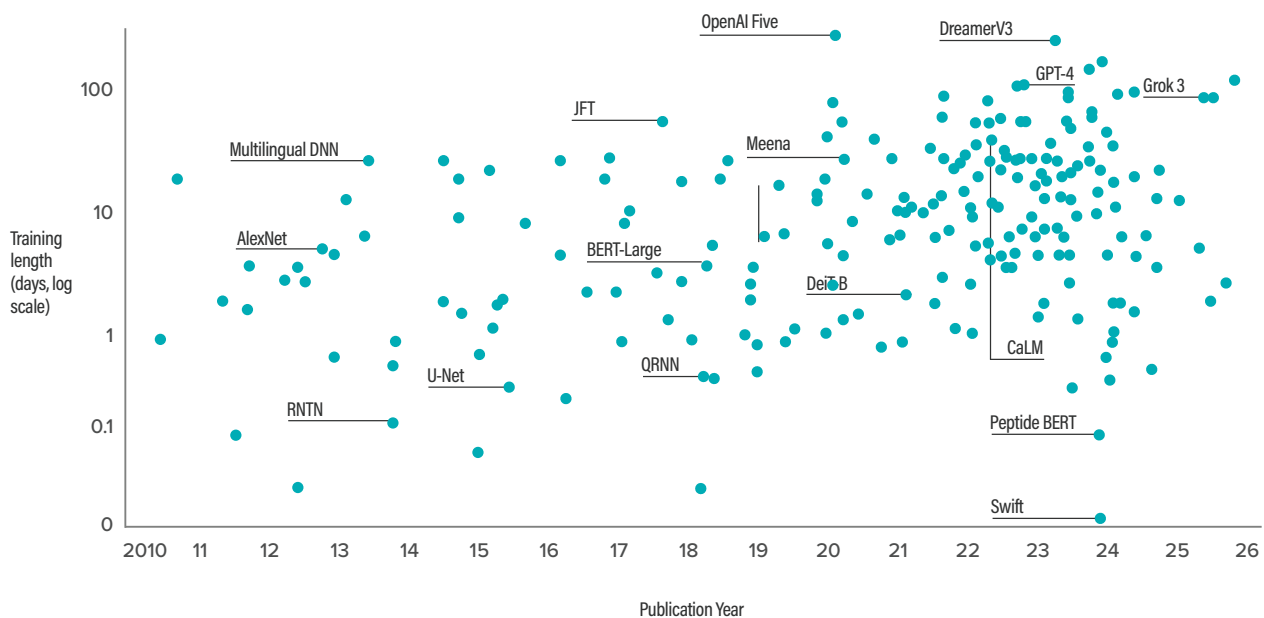
A. How Is AI Software Different?

While traditional algorithms like sorting have clear “correct” outputs we can verify, AI systems operate differently. The energy efficiency principles remain the same—better algorithms use less computation—but measuring “better” becomes far more complex. AI workflows do not exactly fit this paradigm because we do not yet have an equivalent definition of what the “correct” output of an AI algorithm should be. Despite this, there is intense research and development activity seeking to improve the computational efficiency of AI workflows, such as DeepSeek’s innovations with its V3 and R1 systems.

AI systems operate in two phases: training (building the software system) and inference (using the software system). Training modern AI systems increasingly requires larger computational tasks, distributed over tens of thousands of servers, sometimes across geographies (possibly multiple data centers) and often spread over time. Recent state-of-the-art models, such as GPT-4 and Llama 3.1 were trained in dedicated data centers and are estimated to have taken around 90 days of training (see Figure 2.2-1).

This scale and complexity of AI workflows creates unique efficiency challenges. Unlike the sorting example, in which one approach was objectively 100x faster, AI efficiency requires new metrics and measurement approaches.

Figure 2.2-1. Training length of notable AI models, 2010-present



Reproduction of Figure 1.3-14 from Stanford HAI 2025 AI Index Report. Note: y-axis is in logarithmic scale. Training frontier models takes weeks to months, but this time is not necessarily directly related to the total amount of computation because of differences in IT equipment (see Chapter 2.1). Source: Epoch AI | Chart: 2025 AI Index report

B. What Does Efficiency Mean for AI Systems?

AI systems work by processing data using models, which are mathematical frameworks for identifying patterns in data.¹ Unlike in traditional software, we cannot verify the “correctness” of AI systems. Instead, software efficiency revolves around researching and developing new model designs and computation algorithms. These efforts are anchored around two model characteristics: size and benchmark performance. (For simplicity, this section focuses on large language models and omits audio, image, video and protein models, but similar notions apply.)

AI model size typically refers to the number of parameters—the mathematical values adjusted during training. Modern language models contain billions or even trillions of parameters, with larger models requiring more computational resources for both training and inference. Model size directly impacts memory and processor requirements, as well as energy consumption.

AI model benchmarks serve as standardized tests that allow researchers to measure and compare AI model performance across different tasks and capabilities. These evaluation frameworks assess models on specific domains like language understanding, mathematical reasoning, coding ability or factual knowledge, providing quantitative scores that enable systematic comparison between different models and training algorithms. While benchmarks do not enable true verification of accuracy (the way we can verify whether a list of numbers is indeed correctly sorted), they have become essential for tracking progress in AI development.

C. AI Model Training: Is Larger Always Better?

The dominant architecture powering modern AI application is the transformer.² These algorithms process a sequence of data (like text) by learning relationships between its elements (like words). Recent developments have increased the algorithmic efficiency of transformer-based approaches.

State space models (SSMs), like Mamba, process sequences more efficiently than transformers by maintaining a compressed “state” that captures relevant history, which lowers memory use.³ Mixture of experts (MoE) architectures contain multiple specialized sub-networks (“experts”) but activate only a subset for each

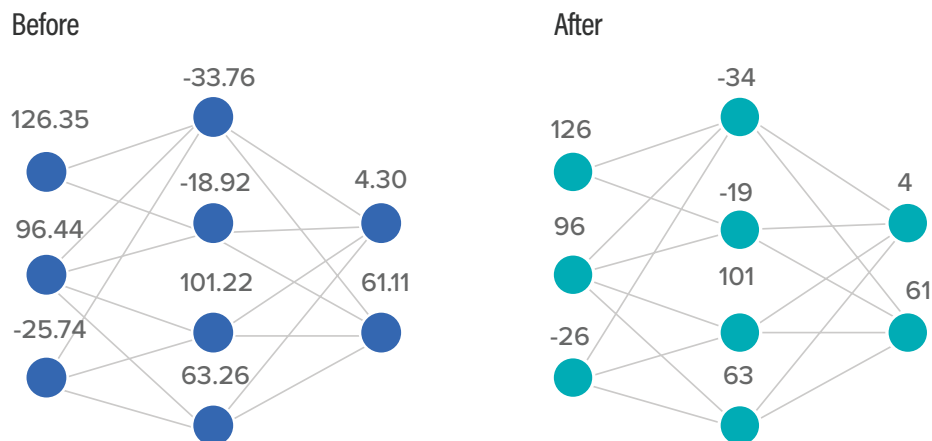
input, dramatically increasing model capacity while keeping computational costs manageable during inference.⁴ While SSMs show promise but limited adoption (notably AI21 Labs' Jamba models combining Mamba with MoE architectures)⁵, MoE architectures have achieved widespread adoption among leading AI labs. GPT-4 is rumored to be MoE-based, as are the recently proposed DeepSeek-v3 and R1 models, while major 2024 releases from Mistral, xAI and Tencent employ MoE architectures.⁶ Modern AI training has evolved beyond simply making models process large datasets; it now focuses on making models learn faster, use less memory, and run more efficiently.

One strategy for increasing computational training speed involves using lower-precision values in training and model construction. Mixed-precision training, for instance, accelerates training by performing operations in half-precision format, which uses fewer bits than the gold standard of 32-bit floating-point representation.⁷ This approach reduces memory requirements and accelerates training while maintaining accuracy by switching to higher-precision numbers only when needed. Quantization, by contrast, speeds up inference by applying the same precision-reduction strategy to the models themselves.⁸ (See Figure 2.2-2.) By switching weights and activation values from 32-bit to 8-bit or even 4-bit formats, quantization can reduce model size by up to 75%, with little to no drop in performance.⁹ Another strategy to improve model efficiency is to downsize the model altogether. Distillation accomplishes this by training smaller models to mimic the behavior of larger ones.

Reinforcement learning enables more efficient training by allowing models to learn optimal reasoning patterns without requiring massive-supervised datasets. DeepSeek R1-Zero demonstrated that purely reinforcement learning-based training from a base model can achieve performance comparable to much larger traditionally trained models, while distilled versions show that sophisticated reasoning capabilities can be compressed into models as small as 1.5B parameters, outperforming conventional models many times their size.¹⁰

Recent innovations in AI architectures inspired by neuroscience and cognitive science have shown promise in outperforming the efficiency of current AI models. Google's Titans, which mimic human memory by combining a neural long-term memory module with attention mechanisms, outperformed both transformers and modern linear recurrent models on long-sequence processing tasks such as language modeling and common-sense reasoning.¹¹ Artificial Kuramoto Oscillatory Neurons (AKOrN) improve tasks such as unsupervised object discovery and adversarial robustness by replicating neuron binding synchronization dynamics.¹²

Figure 2.2-2. Quantization.



These strategies are under active development, and whether they can match the performance of higher-precision models in real-world usage remains under debate. For applications in which accuracy is paramount—such as medical diagnosis or autonomous driving—even minimal performance degradation may be unacceptable, and not all hardware has optimized support for low-precision operations, limiting deployment options.¹³

D. AI Inference: How “Reasoning” Changed Dynamics

While training efficiency focuses on one-time computational costs, inference efficiency affects every use of an AI model. Increasing the efficiency of AI inference can drive significant gains in energy efficiency, making AI deployment more sustainable and cost-effective across diverse applications. However, a recent innovation has complicated this reality beyond what this statement would suggest.

AI “reasoning” models allocate extra computational resources during inference, allowing them to reason through multiple potential responses before selecting the best answer. While this initially appears to increase energy usage, smaller models enhanced with optimized test-time compute can outperform models up to 400x larger that do not use additional computation at test time, ultimately reducing overall energy consumption.¹⁴

Additional innovations promise opportunities for further energy savings. Speculative decoding computes several words in parallel using a smaller “draft” model to predict likely next words, which are then verified by the main model, achieving 2x speed improvements.¹⁵ Pruning large language models removes unnecessary parts to make them smaller and more efficient. Researchers have demonstrated that after removing 20% of the parameters, the pruned model maintains 94.97% of the performance of the original model.¹⁶

This computational redistribution fundamentally alters the energy requirements of AI deployments. Reasoning models have shifted the balance of “expensive training/cheap usage” dramatically: training is a one-time high-cost event, whereas inference costs accumulate over time and can surpass training costs if the model is used extensively.¹⁷ This shift creates a new calculus for AI deployment in which organizations must weigh whether to invest heavily in training massive models that run efficiently or to deploy smaller reasoning-enhanced models that think harder during each query but may ultimately cost more at scale.

E. Reducing Emissions Through Flexible AI Computation

Beyond optimizing individual algorithms, data centers can achieve substantial efficiency gains by strategically timing and placing AI workloads. This operational flexibility represents a different layer of efficiency optimization. To improve both energy efficiency and carbon efficiency, established techniques such as checkpointing and restarting mechanisms, coupled with distributed and flexible computation, play key roles.

i. Checkpointing and restarting

Checkpoint/restart (C/R) technology is a critical foundation for fault tolerance and computational process management. Checkpointing takes snapshots of the system state at intervals, which facilitates recovery from node failures by restoring jobs to the last captured system state. The ability to restart systems from various checkpoints can then be leveraged to break down large computational tasks. Instead of executing long-running batch jobs in one go, tasks can be suspended, redistributed to different data centers, and preemptively scheduled to facilitate load balancing. C/R enables both spatial and temporal compute flexibility by allowing jobs to pause, save state, and resume elsewhere or later without losing completed work.

Originally conceived during the early development of high-performance computing (HPC), C/R technology was designed to solve problems of input/output operation speed, parallelism, and resilience.¹⁸ HPC clusters running large-scale simulations faced

risks from bugs or hardware failures that could erase hours or days of computation. Distributing the computation led to faster results, while checkpointing allowed developers to restart computation from the last snapshot.

Checkpointing mechanisms, once aimed at improving computational throughput and robustness, now have deep implications for spatial and temporal compute flexibility in modern, carbon-aware AI workflows. For AI training workloads—which operate “offline” without user-facing latency (delay between user input and AI-generated output) requirements—checkpointing allows computation to be distributed across low-carbon regions and scheduled during renewable energy peaks.¹⁹ A large language model training job, for example, can run when renewable power is abundant, pause when grid carbon intensity rises, and restart hours later or in a different data center entirely. The system preserves all progress while optimizing for emissions reduction.

Energy efficiency, which can be defined as the amount of useful computation per watt, is necessary but not sufficient. Carbon efficiency, by contrast, depends not only on how much energy is used but also on what kind of energy powers computation and whether it yields higher emissions elsewhere on the grid.

ii. Spatial and temporal compute flexibility

Computation can be operationally distributed across different locations, taking advantage of variation in energy cost, carbon intensity and infrastructure efficiency. For instance, electricity grids in regions like the hydropower-heavy Pacific Northwest or wind-rich Northern Europe offer lower carbon footprints per unit of compute. Certain data centers are more energy efficient than others due to newer information technology (IT) equipment (see Chapter 2.1), optimized cooling systems (Chapter 2.3), or overall design maturity. Some locations may have surplus renewable energy that would otherwise be curtailed. By dynamically routing jobs to these locations, especially for workloads like AI training that do not require immediate user feedback, systems can reduce their net carbon emissions significantly without sacrificing performance.

Time-based flexibility aligns computation with renewable energy availability on the grid. This is especially effective in regions with predictable renewable peaks, such as solar energy in California, which is abundant during the daytime. Some academic and hyperscale data centers already schedule jobs to run during daylight hours to capitalize on this pattern. Unlike inference or real-time services, training workloads are throughput-driven, making them ideal candidates for time-shifted execution.

The implementation of compute flexibility differs significantly between operational compute-shifting and strategic data center siting. Operational compute-shifting involves moving workloads between existing facilities through automated systems that make sub-hourly decisions based on real-time grid conditions and data transfer

costs. These systems must balance the carbon benefits against the latency and bandwidth constraints of moving large datasets, often limiting shifts to workloads with modest data requirements or pre-positioned datasets. In contrast, data center location selection operates on multi-year timescales, incorporating renewable energy potential alongside traditional siting factors like land availability, water resources, air quality permits and network connectivity. Both approaches require a sophisticated understanding of grid marginal emissions—the actual emissions impact of adding or removing load at specific times and locations. Without accounting for marginal emissions rather than average grid mix, compute-shifting efforts risk increasing carbon intensity by inadvertently displacing cleaner generation sources, underscoring the need for real-time emissions data in scheduling decisions.

By contrast, AI inference workloads serve real-time applications like chatbots and recommendation systems that require millisecond response times. With comparatively lower data usage, AI inference workloads have greater spatial flexibility than the training phase, as computations can be transferred from busy data centers to those with more capacity. However, latency constraints demand immediate feedback, which greatly limits the temporal flexibility available to training workloads. Even for tasks with flexibility in response times—such as the “research” activities many leading AI companies offer—inference generally cannot be paused and redistributed elsewhere over a longer period without interrupting user experience. (See Figure 2.2-3.)

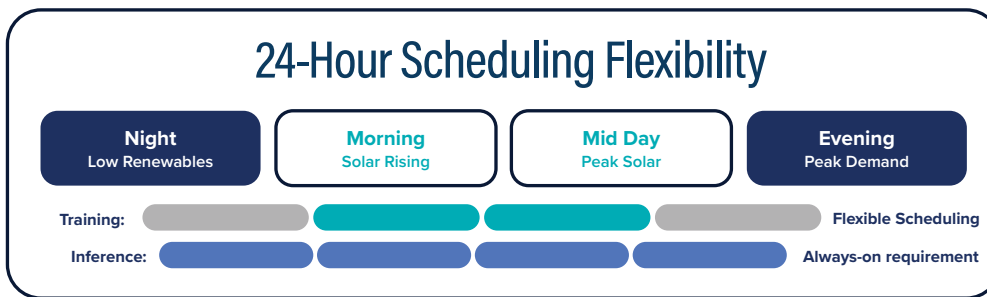
Data centers typically try to maintain constant availability for numerous other latency-sensitive services—video streaming, online banking, e-commerce transactions, email, and gaming—which are similarly more challenging to pause and shift among multiple data centers without disrupting user experience. While training can easily wait for cleaner energy, inference is often expected to run continuously to maintain service availability, thus limiting opportunities for carbon-aware scheduling without degrading user experience. These efficiency techniques, from algorithmic improvements to flexible scheduling, face new challenges as AI applications evolve. Emerging trends like autonomous agents are reshaping the efficiency landscape entirely.

F. An Outlook on AI Software Efficiency

Modern AI systems are increasingly contributing to data center computations. Autonomous agents and coding assistants will likely increase inference costs. Recent AI agent trends are driving substantial inference demand through autonomous workflows, multi-agent systems, and continuous reasoning. Adoption of AI agents for 2% of business tasks is projected to drive a 15% increase in inference compute.²⁰ AI agents are software applications that can independently complete multi-step tasks by using various tools and making decisions along the way. Agents perform extended chains of operations—planning, tool usage, environment interaction, and iterative problem-solving—requiring hundreds of inference calls per task compared with single-

shot responses. Multi-agent frameworks multiply this effect as agents collaborate, negotiate and coordinate through continuous communication.

Figure 2.2-3. Flexibility of training workloads and inference workloads.



Agent workflows are expected to consume up to 100x more inference compute than traditional chatbot interactions.²¹ These workflows drive a combination of always-on operations, where unlike human-initiated queries, agents run continuously for monitoring, scheduling and completing automated tasks. As agents become more capable, organizations that deploy them across more processes will drive growth in compute demand.

Box 2.2-1

Opportunities and challenges in AI energy efficiency research

AI researchers are focusing on three strategies to improve energy efficiency: (1) shorter outputs—compressing lengthy reasoning chains into concise yet effective responses; (2) smaller models—developing compact language models with strong reasoning capabilities through techniques such as knowledge distillation, model compression, and reinforcement learning; and (3) faster responses—designing efficient decoding strategies to accelerate inference.²²

The effectiveness of inference optimizations is highly sensitive to workload geometry, software stacks and hardware accelerators.²³ For example, techniques like speculative decoding are beneficial only at low batch sizes, while mixture-of-experts models sometimes incur higher energy costs despite similar active parameters. Although appropriate application of relevant inference efficiency optimizations can reduce total energy use by up to 73% from unoptimized baselines, naive energy analyses based on floating-point operations or theoretical GPU utilization can significantly underestimate real-world energy consumption.

G. Barriers and Risks

i. Limited expertise pool

AI energy efficiency requires specialized knowledge spanning computer science, energy systems and carbon accounting—a narrow field given the complexity involved in measuring AI efficiency versus traditional algorithms.

ii. Benchmark limitations

Unlike traditional software where correctness is verifiable (as with the sorting algorithm example), AI systems lack clear “correct” outputs. Current benchmarks inadequately measure “higher-order” reasoning tasks, making efficiency comparisons challenging and unreliable.

iii. Development pace outstripping efficiency considerations

Rapid advances (DeepSeek V3/R1, transformer innovations, MoE architectures) may prioritize performance over energy optimization. The 100x increase in agent inference demand could materialize before efficiency solutions are broadly implemented. Model oversimplification risks: Efficiency techniques like quantization and distillation risk removing safety guardrails built into larger models. As in medical and autonomous driving applications, even minimal performance degradation may be unacceptable in critical systems.

iv. Model oversimplification risks

Efficiency techniques like quantization and distillation risk removing safety guardrails built into larger models. As in medical and autonomous driving applications, even minimal performance degradation may be unacceptable in critical systems.

H. Recommendations

1. *Educational institutions should **prepare the next generation of computer scientists and policymakers with the conceptual tools to understand and advance software efficiency.***
 - **Create cross-disciplinary curricula** on algorithmic efficiency, carbon-aware computing and AI systems engineering.
 - **Include energy literacy in computer science and AI degree programs**, covering both micro-level optimizations (e.g., quantization) and macro-level system design (e.g., flexible compute).
 - **Develop policy bootcamps or executive courses for non-technical audiences**, such as civil servants, journalists and business leaders, on emerging AI compute trends and their implications for sustainability.
 - **Fund open-access software efficiency toolkits and benchmarks**, especially those focusing on inference-time efficiency and emissions transparency.
2. *Data center operators should **incorporate software-aware workload management and emissions-aware operations** into core planning.*

- **Partner with utility companies and cloud platforms** to offer real-time grid carbon intensity data and renewable energy forecasts for intelligent job scheduling.
 - **Offer time-delayed training products and pricing schemes** that incentivize shifts to lower-carbon AI training workflows.
 - **Provide application programming interfaces (APIs)** that expose real-time energy and emissions data for AI workloads, enabling software developers to optimize code based on environmental impact.
 - **Adopt software-aware procurement criteria** that favor AI models and systems with verifiable efficiency gains or emissions-conscious design.
3. Software development companies **build efficiency into the core of AI model design, training and deployment.**
- **Develop APIs and platforms that report model energy use**, offering customers transparency and options for greener usage.
 - **Collaborate with academia to publish standardized benchmarks and best practices for evaluating software energy use**, not just performance or accuracy.
4. Regulatory bodies should **establish policies that ensure AI progress aligns with the public interest, energy constraints and climate goals.**
- **Incorporate software efficiency and emissions data disclosure requirements into AI governance frameworks**, especially for high-volume models or widely deployed systems.
 - **Mandate transparent compute and energy reporting** for AI systems procured with public funding or deployed in sensitive sectors (e.g., health, education).
 - **Develop incentives for energy-efficient AI systems**, such as research and development tax credits or public procurement preferences.
5. Educational institutions should **prepare the next generation of computer scientists and policymakers with the conceptual tools to understand and advance software efficiency.**
- **Create cross-disciplinary curricula** on algorithmic efficiency, carbon-aware computing and AI systems engineering.

I. References

1. David Sandalow, Colin McCormick, Alp Kucukelbir, Julio Friedmann, Michal Nachany, Hoesung Lee, Alice Hill, Daniel Loehr, Matthew Wald, Antoine Halff, Ruben Glatt, Philippe Benoit, Kevin Karl et al. Artificial Intelligence for Climate Change Mitigation Roadmap (Second Edition) (ICEF Innovation Roadmap Project, November 2024); <https://doi.org/10.7916/2j4p-nw61> (2024).
2. NVIDIA. Large Language Models Explained (NVIDIA glossary); NVIDIA, Santa Clara, California, <https://www.nvidia.com/en-us/glossary/large-language-models/> (Accessed August 2025).
3. Albert Gu & Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv:2312.00752 (2023). <https://doi.org/10.48550/arXiv.2312.00752>.
4. Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton & Jeff Dean. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. arXiv:1701.06538 (2017). <https://doi.org/10.48550/arXiv.1701.06538>.
5. Chris McKay. TII Unveils Falcon Mamba 7B, A New Open-Source State Space Language Model; Maginative, Boston, Massachusetts, <https://www.maginative.com/article/tii-unveils-falcon-mamba-7b-a-new-open-source-state-space-language-model/> (2024).
6. Asif Razzaq. List of Large Mixture of Experts (MoE) Models: Architecture, Performance, and Innovations in Scalable AI Solutions; Marktechpost Media, Inc., Tustin, California, <https://www.marktechpost.com/2024/11/16/list-of-large-mixture-of-experts-moe-models-architecture-performance-and-innovations-in-scalable-ai-solutions/> (2024).
7. NVIDIA. Train With Mixed Precision (User's Guide | NVIDIA Docs; DA-08617-001_v001); NVIDIA Corporation, Santa Clara, California, <https://docs.nvidia.com/deeplearning/performance/mixed-precision-training/index.html> (2024).
8. Meegle. Quantization Vs Mixed Precision; Meegle, Culver City, California, https://www.meegle.com/en_us/topics/quantization/quantization-vs-mixed-precision#understanding-the-basics-of-quantization-and-mixed-precision (2025).
9. Robert McMenemy. How to 8-bit quantize large models using bits and bytes; AI Accelerator Institute (AI AI), San Francisco, California, <https://www.aiacceleratorinstitute.com/how-to-8-bit-quantize-large-models-using-bits-and-bytes/> (2025).
10. DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948 (2025). <https://doi.org/10.48550/arXiv.2501.12948>.
11. Ali Behrouz, Peilin Zhong & Vahab Mirrokni. Titans: Learning to Memorize at Test Time. arXiv:2501.00663 (2024). <https://doi.org/10.48550/arXiv.2501.00663>.
12. Takeru Miyato, Sindy Löwe, Andreas Geiger & Max Welling. Artificial Kuramoto Oscillatory Neurons. arXiv:2410.13821 (2024). <https://doi.org/10.48550/arXiv.2410.13821>.
13. Ashish Abraham. The Ultimate Handbook for LLM Quantization; Towards Data Science, Insight Media Group, LLC, Toronto, Ontario, Canada, <https://towardsdatascience.com/the-ultimate-handbook-for-llm-quantization-88bb7cb0d9d7/> (2024).
14. Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang & Bowen Zhou. Can 1B LLM Surpass 405B LLM? Rethinking Compute-Optimal Test-Time Scaling. arXiv:2502.06703 (2025). <https://doi.org/10.48550/arXiv.2502.06703>.

15. Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre & John Jumper. Accelerating Large Language Model Decoding with Speculative Sampling. arXiv:2302.01318 (2023). <https://doi.org/10.48550/arXiv.2302.01318>.
16. Xinyin Ma, Gongfan Fang & Xinchao Wang. “LLM-pruner: on the structural pruning of large language models” in Proceedings of the 37th International Conference on Neural Information Processing Systems, New Orleans, LA, USA. 21702-21720 (Article No. 21950), <https://dl.acm.org/doi/10.5555/3666122.3667072> (2023).
17. Kyle Aubrey. How the Economics of Inference Can Maximize AI Value; NVIDIA, Santa Clara, California, <https://blogs.nvidia.com/blog/ai-inference-economics/>
18. Kathryn Mohror Adam Moody, Franck Cappello. SCR Framework: Accelerating Resilience and I/O for Supercomputing Applications (Prepared for the 2019 R&D 100 Award Entry); Lawrence Livermore National Laboratory (LLNL), Livermore, California, https://ipo.llnl.gov/sites/default/files/2020-01/SCR-RD100final_version2.pdf (2019).
19. Andrew A Chien, Liuzixuan Lin, Hai Nguyen, Varsha Rao, Tristan Sharma & Rajini Wijayawardana. “Reducing the Carbon Impact of Generative AI Inference (today and in 2035)” in Proceedings of the 2nd Workshop on Sustainable Computer Systems, Boston, MA, USA. Article 11, <https://doi.org/10.1145/3604930.3605705>, (2023).
20. Ross Sandler Tom O’Malley, Trevor Young. The next wave of AI: Demand and adoption; Barclays, London, United Kingdom, <http://www.ib.barclays/our-insights/3-point-perspective/the-next-wave-of-AI-demand-and-adoption.html> (2024).
21. Kif Leswing. Nvidia CEO Huang says AI has to do ‘100 times more’ computation now than when ChatGPT was released; CNBC, Englewood Cliffs, New Jersey, <https://www.cnbc.com/2025/02/26/nvidia-ceo-huang-says-next-generation-ai-will-need-more-compute.html> (2025).
22. Sicheng Feng, Gongfan Fang, Xinyin Ma & Xinchao Wang. Efficient Reasoning Models: A Survey. arXiv:2504.10903 (2025). <https://doi.org/10.48550/arXiv.2504.10903>.
23. Jared Fernandez, Clara Na, Vashisth Tiwari, Yonatan Bisk, Sasha Luccioni & Emma Strubell. Energy Considerations of Large Language Model Inference and Efficiency Optimizations. arXiv:2504.17674 (2025). <https://doi.org/10.48550/arXiv.2504.17674>.