Compute Requirements for Algorithmic Innovation in Frontier AI Models

Peter Barnett [1]

Abstract

Algorithmic innovation in the pretraining of large language models has driven a massive reduction in the total compute required to reach a given level of capability. In this paper we empirically investigate the compute requirements for developing algorithmic innovations. We catalog 36 pre-training algorithmic innovations used in Llama 3 and DeepSeek-V3. For each innovation we estimate both the total FLOP used in development and the FLOP/s of the hardware utilized. Innovations using significant resources double in their requirements each year. We then use this dataset to investigate the effect of compute caps on innovation. Our analysis suggests that compute caps alone are unlikely to dramatically slow AI algorithmic progress. Even stringent compute caps—such as capping total operations to the compute used to train GPT-2 or capping hardware capacity to 8 H100 GPUs—could still have allowed for half of the cataloged innovations.

1. Introduction

The control of computing resources is a central lever in governing the development of AI (Sastry et al., 2024). This includes setting training compute thresholds above which training must be reported, monitored, and potentially banned (Heim & Koessler, 2024; Miotti et al., 2024; Aguirre, 2025). Nations may also use control over compute to limit their rivals' AI progress, or to prevent malicious non-state actors from gaining access to dual-use AI systems (Heim et al., 2024; Scher & Thiergart, 2024).

AI capabilities can be increased by spending more compute (Hoffmann et al., 2022) (training larger models on more data) and by using more efficient algorithms. The development of increasingly efficient algorithms is referred to as *algorithmic progress* and is the focus of this paper.

[1]Machine Intelligence Research Institute, CA, USA. Correspondence to: Peter Barnett <peter@intelligence.org>.

Workshop on Technical AI Governance (TAIG) at ICML 2025, Vancouver, Canada.

Continued algorithmic progress may limit the effectiveness of compute governance. If less compute is needed to develop dangerous AI capabilities, it may not be possible for nations to monitor all relevant compute. Currently, the amount of compute needed to reach a given level of capability declines by a factor of approximately 3 each year (Ho et al., 2024).

However, algorithmic progress itself also depends on access to compute; researchers must run experiments to develop and validate algorithmic innovations. Hence restricting compute may slow algorithmic progress.

The term "compute" may be ambiguous; in this paper we discuss both:

- *Total operations*: The number of operations used, measured in FLOP.
- *Hardware capacity*: The number of operations per second on the available hardware, measured in TFLOP/s and determined by the number and type of accelerators (e.g., GPUs, TPUs) used.

Some algorithmic innovations require many total operations to develop, such as when the development requires training large models. Other innovations can require very few total operations, but do require a large hardware capacity. For example, parallelization improvements may not require whole models to be trained, but do require many GPUs to test; these innovations then allow hardware to be used more effectively when training models. We consider a diverse range of innovations beyond merely improving training loss efficiency; we include improvements in hardware utilization, model context length, and various other areas.

Table 1. Examples of cataloged algorithmic innovations and their estimated total operations (FLOP) and hardware capacity (TFLOP/s). Full list in Appendix A.

Innovation	FLOP	TFLOP/s
Byte-pair encoding (Sennrich et al., 2016)	9.00×10^{18}	23
Transformer (Vaswani et al., 2017)	4.00×10^{19}	85
AdamW (Loshchilov & Hutter, 2018)	1.76×10^{19}	329
RMSNorm (Zhang & Sennrich, 2019)	1.37×10^{20}	149
ZeRO (Rajbhandari et al., 2020)	Negligible	50 000
FlashAttention (Dao et al., 2022)	6.22×10^{20}	2540

In this paper, we undertake the first empirical investigation of the relationship between compute and the development of algorithmic innovations used in open frontier models. Examples of these innovations are shown in Table 1. We then investigate the impact of compute caps (restrictions on compute) by looking at how many of these innovations could have been developed under various restrictions.

2. Methodology

We catalog pre-training algorithmic innovations implemented in two prominent open models: Llama 3 (Grattafiori et al., 2024) and DeepSeek-V3 (Liu et al., 2024b). We focus on innovations which were actually implemented, as opposed to highly cited. We include innovations that "inspired" modifications if these are cited.

Llama 3 is based on Llama 2 (Touvron et al., 2023b) which in turn was based on LLaMA 1 (Touvron et al., 2023a), DeepSeek-V3 is largely based on DeepSeek-V2 (Liu et al., 2024a), and so we focus on the algorithmic innovations implemented in each of these five models. These models were chosen because they were state-of-the-art for open model capabilities when released, and the developers have provided extensive details about their training.

For each innovation, we review the original paper, and calculate the total operations used for the experiments in the paper, and the total terraFLOP/s (TFLOP/s) of the hardware used (the hardware capacity). Calculating total operations primarily involves calculating the compute used to train each model trained in the paper, and then summing these values. Some compute values were obtained from personal correspondence with paper authors. In various innovations, the compute used was negligible, or the original paper did not provide sufficient information to determine the FLOP or TFLOP/s.

For example, to estimate compute for the original Transformer architecture (Vaswani et al., 2017), we calculate and sum the total operations used to train all models trained as part of the paper, including ablation studies. The paper also states that all models were trained on one machine with 8 NVIDIA P100 GPUs.

We additionally classify the innovations into categories:

- Architecture: Modifications to model architecture (e.g., layers, activations, position embeddings).
- Data & Tokenization: Methods for data processing, tokenization, or selection.
- Efficiencies: Techniques reducing cost (FLOP, memory, time) per step (e.g., low precision, faster algorithms).
- Scaling laws: Experiments to determine how to allocate compute while scaling models.
- Scaling training: Techniques for efficient large-scale distributed training (e.g., parallelism strategies).
- Training: Changes to the training process (e.g., optimizers, loss functions).

2.1. Negligible FLOP Innovations

Various innovations are more efficient implementations of mathematically equivalent operations (Rabe & Staats, 2021; Korthikanti et al., 2023; Dao et al., 2022). These innovations do not require large training runs to validate, because the result would be mathematically identical, but do require significant hardware capacity to be tested at scale (e.g., with a relatively small number of forward and backward passes). 9 out of 36 (25%) of the innovations cataloged used negligible FLOP.

3. Trends

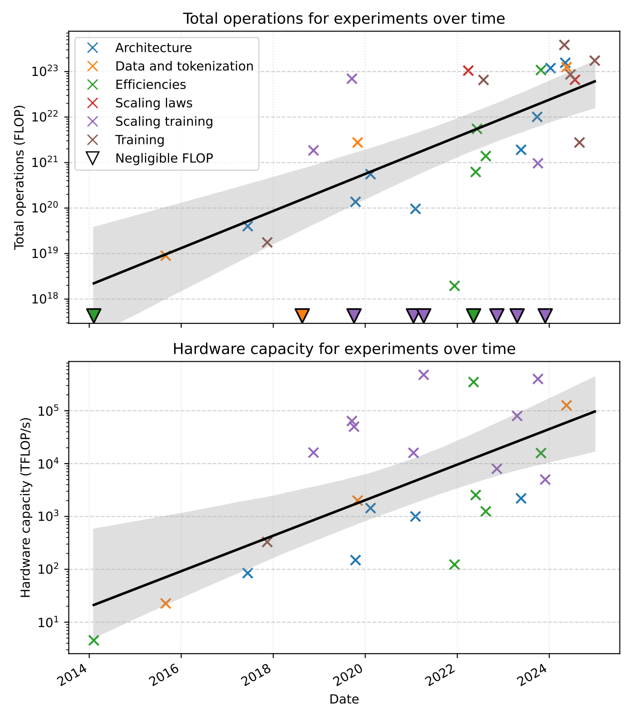

Figure 1. Top: The total operations (in FLOP) for the different algorithmic innovations over time, colored by their category. The triangles indicate innovations which used negligible FLOP in their experiments. The black line shows the trend, and the shaded area indicates the 95% CI for this trend. This trend line is only for innovations that did not use negligible compute.
Bottom: The hardware capacity (in TFLOP/s) for the different algorithmic innovations. The black line shows the trend, and the shaded area indicates the 95% CI for this trend.

We first investigate how the compute requirements for algorithmic innovations are changing over time (Figure 1).

For experiments which do not use negligible compute, the FLOP used is increasing at a rate of $\times 2.53$/year (95% CI: 1.86–3.38). The hardware capacity in TFLOP/s is increasing at a rate of $\times 2.14$/year (95% CI: 1.44–2.76). Pre-training compute performance per dollar (measured in TFLOP/s per dollar) is increasing at a rate of around $\times 1.3$/year (Epoch AI, 2024a). Both the total operations for innovations and the hardware capacity are increasing faster than this, implying that the increased compute use is not due purely to cheaper compute.

The increase in FLOP used for experiments is likely at least partially due to increased hardware capacity. Figure 2 shows that the total operations used in the experiments for innovations is correlated with hardware capacity.

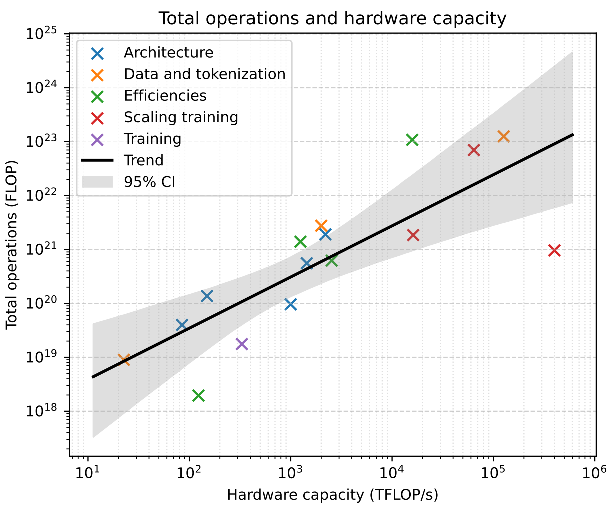

Figure 2. Total operations used for experiments versus hardware capacity for the different algorithmic innovations. Compute used is correlated with hardware capacity. This graph only includes innovations which used non-negligible FLOP.

4. Impact of Compute Caps

Compute caps may reduce the rate of algorithmic progress. Along with the direct impact of preventing the development of dangerous AI systems, caps may also increase the time it takes until dangerous AI systems can be developed with sub-threshold levels of compute. We consider two implementations of compute caps: restrictions on the total operations (*FLOP caps*), and restrictions on the hardware capacity (*hardware caps*).

FLOP caps would entail capping the total number of computational operations an actor (such as a researcher or AI developer) can use in a period of time. The budget could be reset at regular intervals (e.g., allowing 10^{21} FLOP per month), or after audits confirmed that operations were not being used for prohibited activities. This could work by allocating centralized cloud compute (Heim et al., 2024), or having on-chip mechanisms to count the number of operations and enforce usage limits (Kulp et al., 2024; Petrie et al., 2024).

Alternatively, hardware caps could involve restricting the hardware that actors are allowed to own and use. This could be measured by the combined TFLOP/s of their AI-specific hardware. For example, a hardware cap of 5000 TFLOP/s would allow actors to own and use 5 H100 GPUs or 16 A100 GPUs. Hardware caps could be enabled by implementing location verification mechanisms on AI chips combined with a centralized chip registry (Brass & Aarne, 2024).

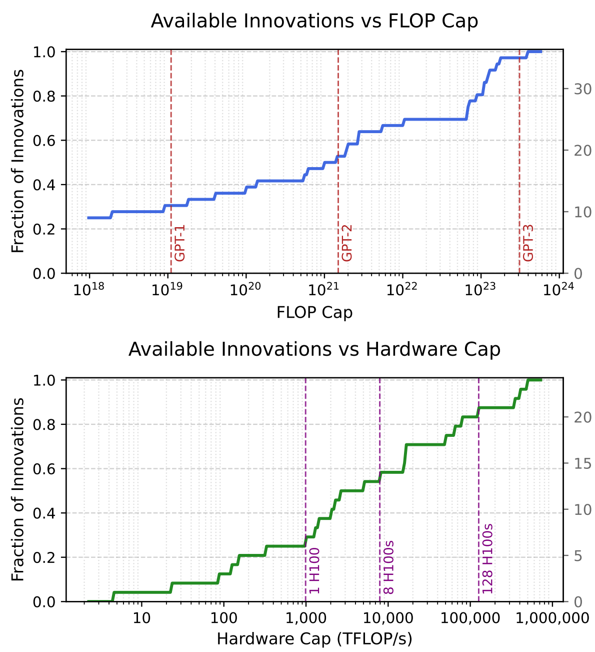

Figure 3. Top: The fraction (left axis) and cumulative count (right axis) of innovations whose total operations falls below a given cap value. Each point on the line answers: "If regulators limited training runs to $\leq X$ FLOP, how many of the techniques in this dataset would still have been discoverable?" For reference, we show the training compute for GPT-1, 2 and 3 (Epoch AI, 2024b). Bottom: Identical analysis for hardware caps. For reference, we show the hardware capacity of different numbers of H100 GPUs. Moving left-to-right shows how the pool of accessible innovations grows as caps loosen.

As a preliminary measure of the impact of compute caps on algorithmic progress, we calculate what fraction of the cataloged innovations fall below a given FLOP or hardware cap (Figure 3). That is: what fraction of the innovations could have been developed if a compute cap had been enforced at a certain level? This estimate will overstate the effects of compute caps, as researchers could attempt to be more efficient with the total operations that they are allocated or run their hardware for longer. However, the x-axes of these plots are on a log-scale, and so even doubling the effective compute cap would have a relatively small effect.

The results in Figure 3 show that **compute caps alone are unlikely to dramatically slow AI algorithmic progress**. Even very restrictive caps such as capping total operations to the compute used to train GPT-2 or capping hardware capacity to 8 H100s would both only disallow about half of the cataloged innovations.

A further implication is that US-led export controls on AI hardware are unlikely to slow rates of AI algorithmic progress in rival states. The existing hardware in these states is likely sufficient for current algorithmic innovation.

As observed in Figure 1, the total operations and hardware capacity are growing over time; if these trends continue to 2028, the median hardware capacity would be 10^6 TFLOP/s (around 1000 H100 GPUs) and the median total operations for innovations using non-negligible FLOP would be 10^{24} FLOP. Compute caps at these levels may therefore slow AI algorithmic progress. These caps are would still be fairly low, as they are 10–100× lower than the hardware and total operations used to train frontier AI models in 2025.

5. Limitations and Future Work

While this study provides initial estimates for the compute requirements of algorithmic innovations, there are several limitations inherent in our methodology and scope. We also propose avenues for future research to build upon this work.

5.1. Limitations

Focus on Reported Successes: Our analysis relies on the experiments described in the final publications introducing an innovation. This approach misses the potentially large amount of compute spent on unsuccessful experiments and preliminary explorations that did not make it into the published results. The compute costs reported in this paper are a *lower bound* on the actual compute used. Although, because researchers could likely be more efficient with their compute, these are not lower bounds on the compute *required* to develop these innovations.

Focus on Initial Research: We focus on estimating compute used in the initial research introducing an innovation. However, validating innovations at scale may often require much more compute than is used initially, and only some cataloged papers include large-scale validation. Our estimates may therefore miss follow-up work (often spread across multiple papers) that demonstrates an innovation continues to work with scale.

Ignoring Research Lineage: We estimate the compute associated with the specific paper introducing or validating an innovation. This does not account for the cumulative compute of the prior work upon which that innovation builds.

Exclusion of Proprietary Models: Our reliance on open models means we miss innovations developed and implemented in closed, proprietary systems (e.g., at companies like Google DeepMind, OpenAI, Anthropic). These labs often operate at the largest scales, and their internal innovations could have different compute requirements.

Pre-training Focus: We deliberately restricted our scope to pre-training innovations. Substantial algorithmic progress also occurs in post-training techniques (e.g., instruction fine-tuning (Ouyang et al., 2022), RLHF (Stiennon et al., 2020), reinforcement learning for reasoning capabilities (Jaech et al., 2024; Guo et al., 2025)), whose compute requirements are not analyzed in this paper. Currently, post-training generally uses much less compute than pre-training (Davidson et al., 2023), so it may also be that experiments to develop algorithmic innovations for pre-training require significantly more compute than for post-training innovations. However, this may not continue to hold as increasing compute is spent on reinforcement learning.

Data Availability and Reporting Bias: The estimation of total operations and hardware capacity relies on the details provided in the source papers. This information is not always available or reported consistently, leading to missing data points. There may be a selection effect where certain researchers are more likely or able to report the hardware used, potentially biasing our view of typical requirements. This includes research from industry developers, who may not want to disclose details about their companies' hardware capabilities. Our estimates are approximations based on available data.

Citation Bias: Developers are more likely to cite work and innovations from their own organization. For example, 5 out of 17 innovations cataloged for DeepSeek-V3 were developed by DeepSeek. Because of this, the proportion of Chinese innovations in Appendix A is likely overstated.

Compute Abundance and Scarcity: This analysis examines past trends under conditions of relative compute abundance for leading labs. It does not model how research might adapt if compute were restricted. Under this regime, researchers would be incentivized to use their compute more efficiently. Under conditions of relative scarcity, researchers would likely run experiments at smaller scales and halt experiments earlier if they did not show promising results. This would likely also change the type of innovations being developed in order to respond to the restrictions ("scarcity breeds creativity"). It is plausible that strong compute caps could incentivize a shift towards innovations that require less compute to develop, and are more useful for AI systems which require less compute to train.

5.2. Future Work

This paper demonstrates the feasibility and potential utility of analyzing the compute requirements for algorithmic innovation. This is a tractable and important research direction for understanding the dynamics of AI development and informing governance. Future work could expand and refine this analysis in several ways:

Qualitative Validation: Complement the quantitative analysis derived from papers with qualitative methods. Interviewing AI researchers could provide valuable insights into the ratio of reported compute versus the actual R&D compute (including failed experiments), helping to calibrate the estimates presented here.

Broader Model and Technique Coverage: Extend the analysis to include other open model series, such as the OLMo models (Groeneveld et al., 2024; OLMo Team et al., 2024). Crucially, expand the scope beyond pre-training to include post-training innovations, including the post-training innovations used in the models covered in this paper, as well as other models such as DeepSeek-R1 (Guo et al., 2025) and Tülu 3 (Lambert et al., 2024).

Focus on Distributed Training: Investigate the specific compute costs associated with developing novel *distributed training* algorithms (e.g., methods like DiLoCo (Douillard et al., 2023) and those used in INTELLECT-1 (Jaghouar et al., 2024)). Progress in distributed training could significantly challenge compute governance regimes aimed at limiting training above a certain scale, making the R&D compute for these techniques important to understand.

Quantifying Innovation Impact (CEG): Connect the *cost* of developing an innovation (as estimated here) with how useful that innovation was. Future work could attempt to estimate a form of Compute-Equivalent Gain (CEG) (Davidson et al., 2023) specifically attributable to different algorithmic innovations, allowing analysis of the "return on compute investment" for different types of algorithmic progress.

6. Impact Statement

This research focuses on a key challenge for compute governance: algorithmic progress may eventually allow actors to circumvent restrictions on computational resources, and gain access to dangerously capable AI systems. If algorithmic progress continues at its current pace, compute governance strategies could become ineffective. We hope this work encourages more researchers to investigate this relationship and other critical questions in compute governance.

Our analysis of compute caps suggests that only very tight restrictions would dramatically slow algorithmic progress. However, we caution against interpreting these results as evidence that algorithmic progress cannot be managed. Rather, compute caps likely need to be combined with other approaches, such as legal frameworks, to be effective for this aim.

We recognize that limiting algorithmic innovation involves tradeoffs, as restrictions would delay both the risks and benefits of AI advancement. Our primary concern is that unchecked algorithmic progress could undermine governance mechanisms designed for responsible AI development. It is likely worth trading some speed in AI progress in exchange for safer and less destabilizing development.

References

Aguirre, A. Keep the Future Human: Why and How We Should Close the Gates to AGI and Superintelligence, and What We Should Build Instead. *arXiv preprint arXiv:2311.09452*, 2025. URL `https://arxiv.org/abs/2311.09452v4`.

Ainslie, J., Lee-Thorp, J., Jong, M. d., Zemlyanskiy, Y., Lebron, F., and Sanghai, S. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. December 2023. URL `https://openreview.net/forum?id=hmOwOZWzYE`.

Bauer, M., Treichler, S., and Aiken, A. Singe: leveraging warp specialization for high performance on GPUs. *SIGPLAN Not.*, 49(8):119–130, February 2014. ISSN 0362-1340. doi: 10.1145/2692916.2555258. URL `https://doi.org/10.1145/2692916.2555258`.

Bavarian, M., Jun, H., Tezak, N., Schulman, J., McLeavey, C., Tworek, J., and Chen, M. Efficient Training of Language Models to Fill in the Middle, July 2022. URL `http://arxiv.org/abs/2207.14255`. arXiv:2207.14255 [cs].

Brass, A. and Aarne, O. Location verification for ai chips. Issue brief, Institute for AI Policy and Strategy (IAPS), April 2024. URL `https://www.iaps.ai/research/location-verification-for-ai-chips`.

Dai, D., Deng, C., Zhao, C., Xu, R., Gao, H., Chen, D., Li, J., Zeng, W., Yu, X., Wu, Y., et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and Ré, C. FLASHATTENTION: fast and memory-efficient exact attention with IO-awareness. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pp. 16344–16359, Red Hook, NY, USA, November 2022. Curran Associates Inc. ISBN 978-1-71387-108-8.

Davidson, T., Denain, J.-S., Villalobos, P., and Bas, G. Ai capabilities can be significantly improved without expensive retraining. *arXiv preprint arXiv:2312.07413*, 2023.

DeepSeek-AI, Zhu, Q., Guo, D., Shao, Z., Yang, D., Wang, P., Xu, R., Wu, Y., Li, Y., Gao, H., Ma, S., Zeng, W., Bi, X., Gu, Z., Xu, H., Dai, D., Dong, K., Zhang, L., Piao, Y., Gou, Z., Xie, Z., Hao, Z., Wang, B., Song, J., Chen, D., Xie, X., Guan, K., You, Y., Liu, A., Du, Q., Gao, W., Lu, X., Chen, Q., Wang, Y., Deng, C., Li, J., Zhao, C., Ruan, C., Luo, F., and Liang, W. DeepSeek-Coder-V2: Breaking the Barrier of Closed-Source Models in Code Intelligence, June 2024. URL http://arxiv.org/abs/2406.11931. arXiv:2406.11931 [cs].

Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pp. 30318–30332, Red Hook, NY, USA, November 2022. Curran Associates Inc. ISBN 978-1-71387-108-8.

Douillard, A., Feng, Q., Rusu, A. A., Chhaparia, R., Donchev, Y., Kuncoro, A., Ranzato, M., Szlam, A., and Shen, J. Diloco: Distributed low-communication training of language models. *arXiv preprint arXiv:2311.08105*, 2023.

Epoch AI. Data on machine learning hardware", 10 2024a. URL https://epoch.ai/data/machine-learning-hardware. Accessed: 2025-04-28.

Epoch AI. Data on notable ai models, 6 2024b. URL https://epoch.ai/data/notable-ai-models. Accessed: 2025-05-11.

Gloeckle, F., Idrissi, B. Y., Rozière, B., Lopez-Paz, D., and Synnaeve, G. Better & faster large language models via multi-token prediction. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pp. 15706–15734, Vienna, Austria, July 2024. JMLR.org.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models, 2024.

Groeneveld, D., Beltagy, I., Walsh, P., Bhagia, A., Kinney, R., Tafjord, O., Jha, A. H., Ivison, H., Magnusson, I., Wang, Y., et al. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*, 2024.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Heim, L. and Koessler, L. Training compute thresholds: Features and functions in ai regulation. *arXiv preprint arXiv:2405.10799*, 2024.

Heim, L., Fist, T., Egan, J., Huang, S., Zekany, S., Trager, R., Osborne, M. A., and Zilberman, N. Governing through the cloud: The intermediary role of compute providers in ai regulation. *arXiv preprint arXiv:2403.08501*, 2024.

Ho, A., Besiroglu, T., Erdil, E., Owen, D., Rahman, R., Guo, Z. C., Atkinson, D., Thompson, N., and Sevilla, J. Algorithmic progress in language models, 2024.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., Driessche, G. v. d., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. Training Compute-Optimal Large Language Models, March 2022. URL http://arxiv.org/abs/2203.15556. arXiv:2203.15556 [cs].

Huang, Y., Cheng, Y., Bapna, A., Firat, O., Chen, M. X., Chen, D., Lee, H., Ngiam, J., Le, Q. V., Wu, Y., and Chen, Z. GPipe: efficient training of giant neural networks using pipeline parallelism. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, number 10, pp. 103–112. Curran Associates Inc., Red Hook, NY, USA, December 2019.

Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-Kishky, A., Low, A., Helyar, A., Madry, A., Beutel, A., Carney, A., et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Jaghouar, S., Ong, J. M., Basra, M., Obeid, F., Straube, J., Keiblinger, M., Bakouch, E., Atkins, L., Panahi, M., Goddard, C., et al. Intellect-1 technical report. *arXiv preprint arXiv:2412.01152*, 2024.

Korthikanti, V. A., Casper, J., Lym, S., McAfee, L., Andersch, M., Shoeybi, M., and Catanzaro, B. Reducing activation recomputation in large transformer models. *Proceedings of Machine Learning and Systems*, 5:341–353, 2023.

Kudo, T. and Richardson, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In Blanco, E. and Lu, W. (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012/.

Kulp, G., Gonzales, D., Smith, E., Heim, L., Puri, P., Vermeer, M. J. D., and Winkelman, Z. *Hardware-Enabled Governance Mechanisms: Developing Technical Solutions to Exempt Items Otherwise Classified Under Export Control Classification Numbers 3A090 and 4A090*. RAND Corporation, Santa Monica, CA, 2024. doi: 10.7249/WRA3056-1.

Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H., Brahman, F., Miranda, L. J. V., Liu, A., Dziri, N., Lyu, S., et al. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Lamy-Poirier, J. Breadth-First Pipeline Parallelism. *Proceedings of Machine Learning and Systems*, 5:48–67, March 2023. URL https://proceedings.mlsys.org/paper_files/paper/2023/hash/24e845415c1486dd2d582a9d639237f9-Abstract-mlsys2023.html.

Li, J., Fang, A., Smyrnis, G., Ivgi, M., Jordan, M., Gadre, S. Y., Bansal, H., Guha, E., Keh, S. S., Arora, K., et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024.

Liu, A., Feng, B., Wang, B., Wang, B., Liu, B., Zhao, C., Dengr, C., Ruan, C., Dai, D., Guo, D., et al. DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model, 2024a.

Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al. DeepSeek-V3 Technical Report, 2024b.

Liu, H., Zaharia, M., and Abbeel, P. RingAttention with Blockwise Transformers for Near-Infinite Context. October 2023. URL https://openreview.net/forum?id=WsRHpHH4s0.

Loshchilov, I. and Hutter, F. Decoupled Weight Decay Regularization. September 2018. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Miotti, A., Bilge, T., Kasten, D., and Newport, J. A Narrow Path, December 2024. URL https://www.narrowpath.co/.

Narayanan, D., Shoeybi, M., Casper, J., LeGresley, P., Patwary, M., Korthikanti, V., Vainbrand, D., Kashinkunti, P., Bernauer, J., Catanzaro, B., Phanishayee, A., and Zaharia, M. Efficient large-scale language model training on GPU clusters using megatron-LM. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–15, St. Louis Missouri, November 2021. ACM. ISBN 978-1-4503-8442-1. doi: 10.1145/3458817.3476209. URL https://dl.acm.org/doi/10.1145/3458817.3476209.

Noune, B., Jones, P., Justus, D., Masters, D., and Luschi, C. 8-bit Numerical Formats for Deep Neural Networks, June 2022. URL http://arxiv.org/abs/2206.02915. arXiv:2206.02915 [cs].

OLMo Team, Walsh, P., Soldaini, L., Groeneveld, D., Lo, K., Arora, S., Bhagia, A., Gu, Y., Huang, S., Jordan, M., et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Peng, H., Wu, K., Wei, Y., Zhao, G., Yang, Y., Liu, Z., Xiong, Y., Yang, Z., Ni, B., Hu, J., Li, R., Zhang, M., Li, C., Ning, J., Wang, R., Zhang, Z., Liu, S., Chau, J., Hu, H., and Cheng, P. FP8-LM: Training FP8 Large Language Models, December 2023. URL http://arxiv.org/abs/2310.18313. arXiv:2310.18313 [cs].

Petrie, J., Aarne, O., Ammann, N., and Dalrymple, D. Interim report: Mechanisms for flexible hardware-enabled guarantees. Technical report, 8 2024.

Qi, P., Wan, X., Huang, G., and Lin, M. Zero Bubble (Almost) Pipeline Parallelism. October 2023. URL https://openreview.net/forum?id=tuzTN0eIO5.

Rabe, M. N. and Staats, C. Self-attention does not need $o(n^2)$ memory, 2021.

Rajbhandari, S., Rasley, J., Ruwase, O., and He, Y. ZeRO: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20, pp. 1–16, Atlanta, Georgia, November 2020. IEEE Press. ISBN 978-1-72819-998-6.

Ren, J., Rajbhandari, S., Aminabadi, R. Y., Ruwase, O., Yang, S., Zhang, M., Li, D., and He, Y. ZeRO-Offload: Democratizing Billion-Scale Model Training. January 2021. URL https://openreview.net/forum?id=qXFQtGMHRa.

Sastry, G., Heim, L., Belfield, H., Anderljung, M., Brundage, M., Hazell, J., O'Keefe, C., Hadfield, G. K., Ngo, R., Pilz, K., et al. Computing power and the governance of artificial intelligence. *arXiv preprint arXiv:2402.08797*, 2024.

Scher, A. and Thiergart, L. Mechanisms to Verify International Agreements About AI Development, November 2024. URL https://techgov.intelligence.org/research/mechanisms-to-verify-international-agreements-about-ai-development.

Sennrich, R., Haddow, B., and Birch, A. Neural Machine Translation of Rare Words with Subword Units. In Erk, K. and Smith, N. A. (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162/.

Shazeer, N. GLU Variants Improve Transformer, February 2020. URL http://arxiv.org/abs/2002.05202. arXiv:2002.05202 [cs].

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism, March 2020. URL http://arxiv.org/abs/1909.08053. arXiv:1909.08053 [cs].

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, February 2024. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.127063. URL https://www.sciencedirect.com/science/article/pii/S0925231223011864.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. volume 30, 2017.

Wang, L., Gao, H., Zhao, C., Sun, X., and Dai, D. Auxiliary-Loss-Free Load Balancing Strategy for Mixture-of-Experts. October 2024. URL https://openreview.net/forum?id=y1iU5czYpE.

Wenzek, G., Lachaux, M.-A., Conneau, A., Chaudhary, V., Guzmán, F., Joulin, A., and Grave, E. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*, 2019.

Xiong, W., Liu, J., Molybog, I., Zhang, H., Bhargava, P., Hou, R., Martin, L., Rungta, R., Sankararaman, K. A., Oguz, B., Khabsa, M., Fang, H., Mehdad, Y., Narang, S., Malik, K., Fan, A., Bhosale, S., Edunov, S., Lewis, M., Wang, S., and Ma, H. Effective Long-Context Scaling of Foundation Models. In Duh, K., Gomez, H., and Bethard, S. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4643–4663, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.260. URL https://aclanthology.org/2024.naacl-long.260/.

Zhang, B. and Sennrich, R. Root mean square layer normalization. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, number 1110, pp. 12381–12392. Curran Associates Inc., Red Hook, NY, USA, December 2019.

Zhao, Y., Gu, A., Varma, R., Luo, L., Huang, C.-C., Xu, M., Wright, L., Shojanazeri, H., Ott, M., Shleifer, S., Desmaison, A., Balioglu, C., Damania, P., Nguyen, B., Chauhan, G., Hao, Y., Mathews, A., and Li, S. PyTorch FSDP: Experiences on Scaling Fully Sharded Data Parallel, September 2023. URL http://arxiv.org/abs/2304.11277. arXiv:2304.11277 [cs].

A. Table of algorithmic innovations

Table 2: Algorithmic Innovations in Large Language Models. ∞ shows innovations used in Llama 3 developed by Meta, 🐋 shows innovations used in DeepSeek-V3. The *Math equiv* column shows innovations which are more computationally efficient ways of performing mathematically equivalent operations.

Innovation	**Date**	**Models**	**FLOP**	**TFLOP/s**	**Category**	**Math equiv**	**Sector**	**Country**
Transformer architecture (Vaswani et al., 2017)	6/12/2017	∞, 🐋	4.00×10^{19}	84.8	Architecture		Industry	USA
RMSNorm (Zhang & Sennrich, 2019)	10/16/2019	∞	1.37×10^{20}	149.42	Architecture		Academic	UK
SwiGLU (Shazeer, 2020)	2/12/2020	∞	5.56×10^{20}	1440	Architecture		Industry	USA
Rotary Position Embedding (Su et al., 2024)	2/3/2021	∞, 🐋	9.62×10^{19}	1000	Architecture		Industry	China
Grouped Query Attention (Ainslie et al., 2023)	5/22/2023	∞	1.91×10^{21}	2200	Architecture		Industry	USA
Increasing RoPE Base Frequency (Xiong et al., 2024)	9/27/2023	∞	1.01×10^{22}	—	Architecture		Industry	USA
DeepSeekMoE (Dai et al., 2024)	1/11/2024	🐋	1.19×10^{23}	—	Architecture		Industry	China
Multi-head Latent Attention (MLA) (Liu et al., 2024a)	5/7/2024	🐋	1.56×10^{23}	—	Architecture		Industry	China
Byte-pair encoding (Sennrich et al., 2016)	8/31/2015	∞, 🐋	9.00×10^{18}	22.58	Data & Tokenization		Academic	UK
SentencePiece (Kudo & Richardson, 2018)	8/19/2018	∞	Negligible	—	Data & Tokenization		Industry	USA
Line-level de-duplication (Wenzek et al., 2019)	11/1/2019	∞	2.76×10^{21}	2000	Data & Tokenization		Industry	USA
Annealing on high quality data (Li et al., 2024)	5/17/2024	∞	1.25×10^{23}	—	Data & Tokenization		Academic	USA

Continued on next page

Table 2: Algorithmic Innovations in Large Language Models – Continued

Innovation	**Date**	**Models**	**FLOP**	**TFLOP/s**	**Category**	**Math equiv**	**Sector**	**Country**
Warp specialization (Bauer et al., 2014)	2/6/2014		Negligible	4.528	Efficiencies	⋆	Academic	USA
Efficient implementation of the causal multi-head attention (Rabe & Staats, 2021)	12/10/2021		1.95×10^{18}	123	Efficiencies	⋆	Industry	USA
Sequence parallelism and selective activation recomputation (Korthikanti et al., 2023)	5/10/2022		Negligible	349 440	Efficiencies	⋆	Industry	USA
FlashAttention (Dao et al., 2022)	5/27/2022		6.22×10^{20}	2539.721	Efficiencies	⋆	Academic	USA
8-bit Numerical Formats for Deep Neural Networks (Noune et al., 2022)	6/6/2022		5.51×10^{21}	—	Efficiencies		Industry	UK
LLM.int8() (Dettmers et al., 2022)	8/15/2022		1.39×10^{21}	1248	Efficiencies		Industry	USA
FP8-LM (Peng et al., 2023)	10/27/2023		1.08×10^{23}	15 824	Efficiencies		Industry	USA
Chinchilla scaling laws (Hoffmann et al., 2022)	3/29/2022		1.05×10^{23}	—	Scaling laws		Industry	UK, USA
LLaMA 3 scaling laws (Grattafiori et al., 2024)	7/23/2024		6.60×10^{22}	—	Scaling laws		Industry	USA
GPipe: Pipeline Parallelism (Huang et al., 2019)	11/16/2018		1.85×10^{21}	16 188.8	Scaling training	⋆	Industry	USA
Tensor parallelism (Shoeybi et al., 2020)	9/17/2019		6.98×10^{22}	64 000	Scaling training	⋆	Industry	USA
Zero Redundancy Optimizer (ZeRO) (Rajbhandari et al., 2020)	10/4/2019	,	Negligible	50 000	Scaling training	⋆	Industry	USA
ZeRO-Offload (Ren et al., 2021)	1/18/2021		Negligible	16 000	Scaling training	⋆	Industry	USA

Continued on next page

Table 2: Algorithmic Innovations in Large Language Models – Continued

Innovation	Date	Models	FLOP	TFLOP/s	Category	Math equiv	Sector	Country
PTD-P (Narayanan et al., 2021)	4/9/2021		Negligible	479 232	Scaling training	⋆	Industry	USA
Breadth-First Pipeline Parallelism (Lamy-Poirier, 2023)	11/11/2022		Negligible	8000	Scaling training	⋆	Industry	Canada
Fully sharded data parallelism (Zhao et al., 2023)	4/21/2023		Negligible	79 872	Scaling training	⋆	Industry	USA
Context Parallelism (Liu et al., 2023)	10/3/2023		9.65×10^{20}	400 000	Scaling training	⋆	Academic	USA
Pipeline Parallelism (PP) (Qi et al., 2023)	11/30/2023		Negligible	4992	Scaling training	⋆	Industry	Singapore
AdamW (Loshchilov & Hutter, 2018)	11/14/2017	,	1.76×10^{19}	329.1	Training		Academic	Germany
Fill-in-Middle (Bavarian et al., 2022)	7/28/2022		6.55×10^{22}	—	Training		Industry	USA
Multi-token Prediction (Gloeckle et al., 2024)	4/30/2024		3.85×10^{23}	—	Training		Industry	USA, France
Fill-in-Middle DeepSeek validation (DeepSeek-AI et al., 2024)	6/17/2024		8.64×10^{22}	—	Training		Industry	China
Auxiliary-loss-free load balancing (Wang et al., 2024)	8/28/2024		2.76×10^{21}	—	Training		Industry	China
Multi-Token Prediction (MTP) (Liu et al., 2024b)	12/27/2024		1.74×10^{23}	—	Training		Industry	China