Review

Breaking the Bottleneck Advances in Efficient Transformer Design

Yawen Bao *

Posted Date: 28 February 2025

doi: 10.20944/preprints202502.2271.v1

Keywords: efficient transformers; self-attention; sparse attention; dimensionality reduction; hierarchical architectures; computational efficiency; deep learning; NLP; vision transformers; hardware optimization

 doi:10.20944/preprints202502.2271.v1

Review

Breaking the Bottleneck Advances in Efficient Transformer Design

Yawen Bao

Department of Computer Science, Tsinghua University, Beijing, China; yawen.bao@tsinghua.edu.cn

Abstract: Transformers have become the backbone of numerous advancements in deep learning, excelling across domains such as natural language processing, computer vision, and scientific modeling. Despite their remarkable performance, the high computational and memory costs of the standard Transformer architecture pose significant challenges, particularly for long sequences and resource-constrained environments. In response, a wealth of research has been dedicated to improving the efficiency of Transformers, resulting in a diverse array of innovative techniques. This survey provides a comprehensive overview of these efficiency-driven advancements. We categorize existing approaches into four major areas: (1) approximating or sparsifying the self-attention mechanism, (2) reducing input or intermediate representation dimensions, (3) leveraging hierarchical and multiscale architectures, and (4) optimizing hardware utilization through parallelism and quantization. For each category, we discuss the underlying principles, representative methods, and the trade-offs involved. We also identify key challenges in the field, including balancing efficiency with performance, scaling to extremely long sequences, addressing hardware constraints, and mitigating the environmental impact of large-scale models. To guide future research, we highlight promising directions such as unified frameworks, dynamic and sparse architectures, energy-aware designs, and cross-domain adaptations. By synthesizing the latest advancements and providing insights into unresolved challenges, this survey aims to serve as a valuable resource for researchers and practitioners seeking to develop or apply efficient Transformer models. Ultimately, the pursuit of efficiency is crucial for ensuring that the transformative potential of Transformers can be realized in a sustainable, accessible, and impactful manner.

Keywords: efficient transformers; self-attention; sparse attention; dimensionality reduction; hierarchical architectures; computational efficiency; deep learning; NLP; vision transformers; hardware optimization

1. Introduction

Transformers have emerged as a dominant architecture in various domains, including natural language processing (NLP), computer vision, and even time-series analysis, due to their exceptional performance in capturing long-range dependencies and contextual information. Since their inception with the introduction of the seminal Transformer model by Vaswani et al. in 2017, transformers have powered a wide array of applications, from language modeling and machine translation to image recognition and protein folding [1]. Despite their widespread success, the original Transformer architecture is computationally expensive, both in terms of memory and processing requirements, particularly when dealing with large-scale data or long sequences. These computational challenges have spurred significant research efforts to make transformers more efficient, giving rise to a diverse set of "efficient transformer" variants [2]. The inefficiency of the standard Transformer model primarily stems from its self-attention mechanism, which computes pairwise interactions between all tokens in a sequence. This computation has a quadratic complexity with respect to the sequence length, making it prohibitively expensive for long sequences [3]. In real-world applications such as document classification, video processing, and large-scale language modeling, where sequences can be thousands

 doi:10.20944/preprints202502.2271.v1

or even millions of tokens long, this quadratic scaling becomes a severe bottleneck [4]. Moreover, the increasing demand for deploying transformer models on resource-constrained devices, such as mobile phones and edge devices, necessitates innovative solutions to reduce the computational and memory requirements of these models while maintaining their performance. To address these challenges, researchers have proposed a variety of approaches to enhance the efficiency of transformers. Broadly, these approaches can be categorized into methods that (1) approximate or sparsify the self-attention mechanism, (2) reduce the dimensionality of the input or intermediate representations, (3) adopt hierarchical or multiscale architectures, and (4) leverage parallelism and hardware-aware optimizations [5]. Each of these strategies aims to strike a balance between computational efficiency and model performance, often tailored to the specific requirements of the target application or deployment environment. Approximation-based methods, for instance, focus on reducing the computational complexity of the self-attention mechanism by approximating the full attention matrix [6]. Techniques such as low-rank factorization, kernel-based approximations, and clustering-based approaches fall into this category [7]. Sparse attention mechanisms, on the other hand, aim to limit the computation to a subset of token interactions, leveraging the observation that many interactions in long sequences are not equally important. Models like Longformer, Reformer, and Big Bird have demonstrated that sparsity can be a powerful tool for reducing computational overhead without significant loss in performance. Dimensionality reduction techniques include methods such as token pruning, pooling, and projection, which reduce the sequence length or embedding size at various stages of the model [8–10]. Hierarchical architectures, inspired by the success of multiscale representations in convolutional neural networks (CNNs), introduce a layered structure where information is processed at varying levels of granularity [11]. These models, such as Swin Transformer and Funnel Transformer, are particularly effective in vision-related tasks, where the hierarchical organization of features aligns well with the underlying structure of images [12]. Parallelism and hardware-aware optimizations represent another frontier in efficient transformer research. Techniques such as model parallelism, pipeline parallelism, and quantization aim to optimize the use of available computational resources [13]. Additionally, frameworks tailored for specific hardware, such as TensorRT and DeepSpeed, have further accelerated the training and inference of transformer models. This survey aims to provide a comprehensive overview of the various techniques and strategies developed to enhance the efficiency of transformers. We categorize and analyze these methods, highlighting their underlying principles, strengths, and limitations. Furthermore, we discuss the trade-offs involved in designing efficient transformer architectures and provide insights into emerging trends and future research directions [2]. By synthesizing the vast and rapidly growing body of literature on efficient transformers, we hope to offer a valuable resource for researchers and practitioners seeking to develop or utilize transformer models in a computationally sustainable manner [14].

2. Background

The Transformer architecture, introduced by Vaswani et al. in 2017, revolutionized deep learning by replacing the recurrent and convolutional mechanisms traditionally used for sequence modeling with a self-attention mechanism [15]. This innovation allowed Transformers to model dependencies across entire sequences in parallel, resulting in significant improvements in both computational efficiency and performance on a variety of tasks. The architecture's core building blocks—multi-head self-attention, feed-forward networks, residual connections, and layer normalization—work together to capture complex relationships between input tokens, making Transformers versatile across a wide range of applications.

2.1. The Standard Transformer Architecture

At the heart of the Transformer is the self-attention mechanism, which computes the importance of each token in a sequence relative to all other tokens. This is achieved through the scaled dot-product attention formula:

 doi:10.20944/preprints202502.2271.v1

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V,$$

where Q, K, and V are the query, key, and value matrices, and d_k is the dimensionality of the keys [16]. Multi-head attention extends this mechanism by projecting the input into multiple subspaces, allowing the model to capture diverse patterns in the data [17]. The outputs from each attention head are concatenated and linearly transformed. In addition to attention, the Transformer employs position-wise feed-forward networks (FFNs) and positional encodings to introduce order information into the sequence. These components are stacked in layers, with residual connections and layer normalization ensuring stable gradient flow and effective training [18]. The Transformer is divided into two main components: the encoder, which processes the input sequence, and the decoder, which generates the output sequence [19]. The encoder-decoder structure makes it suitable for sequence-to-sequence tasks like machine translation [20]. However, many variants of the Transformer, such as BERT and GPT, focus exclusively on the encoder or decoder for specific tasks.

2.2. Computational Challenges of Transformers

Despite their effectiveness, Transformers face significant computational and memory challenges, especially for long sequences [21]. The self-attention mechanism, with its $O(n^2)$ complexity, becomes the primary bottleneck [22]. For a sequence of length n, the quadratic scaling arises from the need to compute pairwise interactions between all tokens. This issue is exacerbated in domains like natural language processing, where sequence lengths can range from hundreds to thousands of tokens, and in computer vision, where 2D image patches are flattened into sequences. Additionally, Transformers require substantial memory to store intermediate representations during training, which limits their scalability on hardware with constrained resources [9]. These issues are further compounded by the high latency associated with the sequential nature of training steps, particularly in autoregressive models where tokens are processed one at a time during generation.

2.3. Applications Driving the Need for Efficiency

The success of Transformers in a variety of domains has heightened the demand for efficient solutions [23]. In NLP, models such as BERT, GPT, and T5 have pushed the boundaries of language understanding and generation, but their deployment often requires significant computational resources [24]. In computer vision, Vision Transformers (ViTs) have challenged the dominance of convolutional neural networks, while in scientific fields, Transformers are being applied to protein structure prediction, drug discovery, and climate modeling [25]. Real-world constraints such as limited hardware resources, energy efficiency, and latency requirements have made the development of efficient Transformers a critical area of research [26]. Furthermore, the proliferation of edge devices and the need for real-time processing in applications like autonomous systems and conversational AI have underscored the importance of designing lightweight and scalable Transformer models. This section establishes the foundational concepts and challenges that underpin the Transformer architecture, setting the stage for a detailed exploration of the techniques developed to address these limitations in subsequent sections.

3. Approaches to Efficiency

The increasing computational demands of Transformers, particularly for long sequences and large-scale datasets, have driven the development of numerous techniques aimed at improving their efficiency. These approaches span a wide range of strategies, each targeting specific aspects of the Transformer architecture. In this section, we categorize these methods into four major groups: (1) approximating or sparsifying the self-attention mechanism, (2) reducing input or intermediate representation dimensions, (3) introducing hierarchical or multiscale architectures, and (4) optimizing hardware utilization through parallelism and quantization. Each category represents a distinct perspective on improving the computational and memory efficiency of Transformer models [27].

 doi:10.20944/preprints202502.2271.v1

3.1. Approximating or Sparsifying the Self-Attention Mechanism

The self-attention mechanism is the most computationally expensive component of the Transformer, with its $O(n^2)$ complexity [28]. To alleviate this burden, researchers have proposed methods to approximate or sparsify the attention matrix:

- **Low-Rank Factorization:** Techniques such as Linformer and Performers approximate the full attention matrix by decomposing it into lower-dimensional representations. These methods leverage low-rank properties of attention matrices, significantly reducing computational overhead.
- **Kernel-Based Approximations:** Kernel methods reformulate self-attention as a series of linear operations, enabling efficient computation [29]. For example, the Performer model introduces random feature approximations to achieve linear-time complexity [30].
- **Sparse Attention:** Models like Longformer and Big Bird adopt sparse attention patterns, where only a subset of token interactions is computed. These patterns include local windows, global tokens, and dilated spans, offering flexibility and scalability for long sequences.
- **Clustering and Grouping:** Methods like Reformer use locality-sensitive hashing to cluster similar tokens, reducing the number of pairwise comparisons [31]. Cluster-based approaches improve efficiency without significantly compromising performance.

3.2. Reducing Input or Intermediate Representations

Another approach to efficiency involves reducing the size of input sequences or intermediate representations [32]. These methods focus on minimizing redundant computations:

- **Token Pruning:** Techniques like Early Exit and token pruning selectively drop tokens that contribute little to the final output, effectively shortening the sequence length during processing.
- **Pooling and Compression:** Models such as Funnel Transformer and PoolFormer compress token representations at different stages, introducing hierarchical structures that reduce computational requirements [33].
- **Dimensionality Reduction:** Reducing the dimensionality of embeddings or intermediate feature maps can also decrease memory usage and computation without significant performance degradation [34].

3.3. Hierarchical and Multiscale Architectures

Inspired by multiscale feature extraction in convolutional neural networks, hierarchical and multiscale architectures have been adapted for Transformers:

- **Hierarchical Transformers:** These models, such as Swin Transformer, process information at multiple levels of granularity, capturing both local and global patterns. This structure is especially effective in computer vision tasks.
- **Progressive Token Merging:** Hierarchical models progressively merge tokens at higher layers, reducing sequence length and focusing computational resources on salient features [35].
- **Multiscale Attention:** Attention mechanisms that operate across multiple scales ensure that global dependencies are captured while maintaining efficiency.

3.4. Optimizing Hardware Utilization

Efficient utilization of hardware resources is another avenue for improving Transformer efficiency [36]. These techniques focus on leveraging parallelism and reducing model size:

- **Model and Pipeline Parallelism:** Approaches like model and pipeline parallelism split computations across multiple devices, enabling the training of larger models while maintaining efficiency [37].
- **Quantization:** Quantization reduces the precision of model parameters, decreasing memory usage and accelerating computation [38]. Techniques such as post-training quantization and quantization-aware training are commonly used.

 doi:10.20944/preprints202502.2271.v1

- **Hardware-Aware Designs:** Custom hardware accelerators and frameworks, such as TensorRT and DeepSpeed, optimize model performance by tailoring computations to specific hardware.

These approaches highlight the diversity of techniques available for improving Transformer efficiency [39]. In the following sections, we delve deeper into each category, exploring the specific models, algorithms, and trade-offs that define these advancements.

4. Challenges and Future Directions

Despite significant progress in improving the efficiency of Transformer models, numerous challenges remain [8]. These challenges stem from the inherent complexity of the Transformer architecture, the diversity of application requirements, and the growing scale of modern datasets and models [40]. Addressing these challenges is critical for enabling the widespread adoption of Transformers in resource-constrained settings and expanding their applicability to emerging domains [41]. In this section, we discuss the key challenges in efficient Transformer research and highlight promising future directions [42].

4.1. Challenges

4.1.1. Trade-offs Between Efficiency and Performance

One of the fundamental challenges in designing efficient Transformers is achieving an optimal balance between computational efficiency and model performance [43]. Many methods that reduce computational complexity, such as low-rank approximations or token pruning, risk degrading model accuracy, particularly on tasks requiring fine-grained understanding or long-range dependencies [16,44,45]. Striking the right trade-off requires careful tuning and task-specific customization, which can be time-consuming and resource-intensive.

4.1.2. Scalability to Extremely Long Sequences

While significant strides have been made in reducing the quadratic complexity of self-attention, scaling Transformers to handle extremely long sequences (e.g., tens of thousands of tokens) remains challenging [46]. Sparse and hierarchical attention mechanisms often rely on domain-specific assumptions, limiting their generalizability. Moreover, efficiently processing sequences of this scale on existing hardware requires further innovations in memory management and parallelization.

4.1.3. Lack of Standard Benchmarks for Efficiency

Evaluating the efficiency of Transformer models is often inconsistent across studies, with researchers using different hardware setups, metrics, and datasets. This lack of standardization makes it difficult to compare methods fairly and determine their practical utility. Establishing widely accepted benchmarks and evaluation protocols is crucial for advancing the field.

4.1.4. Hardware Constraints

While many efficient Transformer techniques are theoretically promising, their real-world deployment is constrained by hardware limitations [47]. For example, methods relying on complex sparse matrix operations may face inefficiencies on GPUs and TPUs, which are optimized for dense matrix computations. Designing algorithms that align with hardware capabilities remains a pressing challenge.

4.1.5. Environmental and Energy Concerns

Training and deploying large-scale Transformer models have substantial environmental and energy costs, raising concerns about their sustainability [48]. Even efficient Transformer variants can be resource-intensive when applied at scale. Addressing these concerns requires not only algorithmic innovations but also broader efforts to promote energy-efficient AI practices [49].

 doi:10.20944/preprints202502.2271.v1

4.2. Future Directions

4.2.1. Unified Frameworks for Efficiency

Developing unified frameworks that combine multiple efficiency techniques—such as sparse attention, token pruning, and quantization—into a cohesive architecture is a promising direction. Such frameworks could provide task-specific adaptability while maintaining high performance and scalability.

4.2.2. Learning Sparse and Dynamic Structures

Dynamic attention mechanisms, where the model learns to adaptively select the most relevant tokens or interactions for a given task, offer significant potential for improving efficiency. Incorporating sparsity in a learnable and data-driven manner could yield models that are both efficient and generalizable across domains [50].

4.2.3. Energy-Aware Model Design

Future research should emphasize energy-aware model design, integrating energy consumption as a key optimization criterion during training and inference [51]. Techniques such as model pruning, quantization, and low-power hardware accelerators could be tailored to achieve this goal.

4.2.4. Cross-Domain Applications

Extending efficient Transformer techniques to emerging domains, such as bioinformatics, robotics, and scientific computing, presents exciting opportunities. These applications often involve unique data structures (e.g., graphs, irregular grids) that require specialized adaptations of Transformer models [52].

4.2.5. Neuroscience-Inspired Architectures

Drawing inspiration from biological systems, particularly the human brain, could inform the design of more efficient and robust Transformer architectures. Techniques like attention prioritization and hierarchical processing have natural analogs in neural mechanisms, suggesting potential avenues for innovation.

4.2.6. Better Integration with Hardware

The co-design of Transformer algorithms and hardware accelerators is an essential direction for maximizing efficiency. Collaborative efforts between algorithm developers and hardware engineers could result in specialized chips and libraries that optimize Transformer computations for real-world deployment [53].

4.2.7. Open-Source Contributions and Collaboration

Community-driven open-source initiatives play a pivotal role in advancing efficient Transformers [54]. Platforms like Hugging Face, TensorFlow, and PyTorch can incorporate efficient architectures, benchmarks, and tools to facilitate widespread experimentation and adoption [55].

4.3. Conclusion

Efficient Transformers represent a rapidly evolving area of research, driven by the need to overcome the computational limitations of the original Transformer architecture [56]. While significant progress has been made, the challenges discussed highlight the complexity and multifaceted nature of this endeavor [57]. By addressing these challenges and pursuing the outlined future directions, the research community can unlock the full potential of Transformers, making them more accessible, sustainable, and impactful across diverse fields [58].

 doi:10.20944/preprints202502.2271.v1

5. Conclusion

Transformers have revolutionized numerous fields, from natural language processing and computer vision to scientific computing and beyond [59]. Their unparalleled ability to model long-range dependencies and contextual relationships has driven transformative advancements across a broad spectrum of applications. However, their high computational and memory costs remain a significant bottleneck, particularly as the scale of modern datasets and model deployments continues to grow. The quest for efficient Transformers has thus become a critical focus in both academic research and industry innovation [60]. This survey has provided a comprehensive overview of the strategies developed to improve the efficiency of Transformer models. We have explored a diverse set of approaches, including approximating or sparsifying the self-attention mechanism, reducing input or intermediate representation dimensions, leveraging hierarchical and multiscale architectures, and optimizing hardware utilization through parallelism and quantization [61]. Each of these techniques offers unique advantages and trade-offs, highlighting the breadth of ingenuity in tackling the challenges posed by Transformer inefficiencies. Despite these advancements, several challenges persist, including the trade-off between efficiency and performance, scalability to extremely long sequences, hardware constraints, and the environmental impact of large-scale model training and deployment. Addressing these challenges requires continued research and collaboration, with a focus on unified frameworks, dynamic and sparse architectures, energy-aware designs, and cross-domain adaptability. Looking forward, the future of efficient Transformers is bright [62]. Emerging trends, such as integrating biological inspirations, co-designing models with hardware accelerators, and fostering community-driven open-source initiatives, promise to drive further innovations. By building on the foundation of current research, the next generation of efficient Transformer models will likely be more adaptable, sustainable, and impactful, paving the way for new breakthroughs across diverse domains.

Efficient Transformers are not just a technical necessity; they are an enabler of broader accessibility and scalability for deep learning technologies. By addressing their computational challenges, we can democratize their benefits, bringing cutting-edge AI capabilities to a wider range of users, devices, and applications. As the field continues to evolve, it remains essential to prioritize both performance and efficiency, ensuring that the transformative potential of Transformers can be realized in an equitable and sustainable manner.

References

1. Akyürek, E.; Schuurmans, D.; Andreas, J.; Ma, T.; Zhou, D. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661* **2022**.
2. Bellman, R. The theory of dynamic programming. *Bulletin of the American Mathematical Society* **1954**, *60*, 503–515.
3. Xie, S.M.; Raghunathan, A.; Liang, P.; Ma, T. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080* **2021**.
4. Garg, S.; Tsipras, D.; Liang, P.S.; Valiant, G. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems* **2022**, *35*, 30583–30598.
5. Child, R.; Gray, S.; Radford, A.; Sutskever, I. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509* **2019**.
6. Garg, S.; Tsipras, D.; Liang, P.; Valiant, G. What Can Transformers Learn In-Context? A Case Study of Simple Function Classes. In Proceedings of the Advances in Neural Information Processing Systems, 2022.
7. Tay, Y.; Dehghani, M.; Bahri, D.; Metzler, D. Efficient Transformers: A Survey, 2022, [arXiv:cs.LG/2009.06732].
8. Pérez, J.; Marinković, J.; Barceló, P. On the turing completeness of modern neural network architectures. *arXiv preprint arXiv:1901.03429* **2019**.
9. Sanford, C.; Hsu, D.; Telgarsky, M. Representational Strengths and Limitations of Transformers. *arXiv preprint arXiv:2306.02896* **2023**.
10. Zniyed, Y.; Nguyen, T.P.; et al. Efficient tensor decomposition-based filter pruning. *Neural Networks* **2024**, *178*, 106393.
11. Kitaev, N.; Kaiser, Ł.; Levskaya, A. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451* **2020**.
12. Alman, J.; Song, Z. Fast attention requires bounded entries. *arXiv preprint arXiv:2302.13214* **2023**.

 doi:10.20944/preprints202502.2271.v1

13. Tay, Y.; Dehghani, M.; Abnar, S.; Shen, Y.; Bahri, D.; Pham, P.; Rao, J.; Yang, L.; Ruder, S.; Metzler, D. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006* **2020**.
14. Dai, D.; Sun, Y.; Dong, L.; Hao, Y.; Ma, S.; Sui, Z.; Wei, F. Why Can GPT Learn In-Context? Language Models Implicitly Perform Gradient Descent as Meta-Optimizers. In Proceedings of the ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models, 2023.
15. Tay, Y.; Bahri, D.; Yang, L.; Metzler, D.; Juan, D.C. Sparse sinkhorn attention. In Proceedings of the International Conference on Machine Learning. PMLR, 2020, pp. 9438–9447.
16. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.V.; Salakhutdinov, R. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860* **2019**.
17. Furst, M.; Saxe, J.B.; Sipser, M. Parity, circuits, and the polynomial-time hierarchy. *Mathematical systems theory* **1984**, *17*, 13–27.
18. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. In Proceedings of the Advances in neural information processing systems, 2020, Vol. 33, pp. 1877–1901.
19. Merrill, W.; Sabharwal, A.; Smith, N.A. Saturated transformers are constant-depth threshold circuits. *Transactions of the Association for Computational Linguistics* **2022**, *10*, 843–856.
20. Kojima, T.; Gu, S.S.; Reid, M.; Matsuo, Y.; Iwasawa, Y. Large Language Models are Zero-Shot Reasoners. In Proceedings of the Advances in Neural Information Processing Systems, 2022.
21. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, 2019, pp. 4171–4186.
22. Hahn, M.; Goyal, N. A Theory of Emergent In-Context Learning as Implicit Structure Induction. *arXiv preprint arXiv:2303.07971* **2023**.
23. Hao, Y.; Angluin, D.; Frank, R. Formal language recognition by hard attention transformers: Perspectives from circuit complexity. *Transactions of the Association for Computational Linguistics* **2022**, *10*, 800–810.
24. Zhang, Z.; Zhang, A.; Li, M.; Smola, A. Automatic Chain of Thought Prompting in Large Language Models. In Proceedings of the The Eleventh International Conference on Learning Representations, 2023.
25. Olsson, C.; Elhage, N.; Nanda, N.; Joseph, N.; DasSarma, N.; Henighan, T.; Mann, B.; Askell, A.; Bai, Y.; Chen, A.; et al. In-context Learning and Induction Heads. *Transformer Circuits Thread* **2022**. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.
26. Nye, M.; Andreassen, A.J.; Gur-Ari, G.; Michalewski, H.; Austin, J.; Bieber, D.; Dohan, D.; Lewkowycz, A.; Bosma, M.; Luan, D.; et al. Show Your Work: Scratchpads for Intermediate Computation with Language Models. In Proceedings of the Deep Learning for Code Workshop, 2022.
27. Gu, A.; Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752* **2023**.
28. Luo, S.; Li, S.; Cai, T.; He, D.; Peng, D.; Zheng, S.; Ke, G.; Wang, L.; Liu, T.Y. Stable, fast and accurate: Kernelized attention with relative positional encoding. In Proceedings of the Advances in Neural Information Processing Systems, 2021, Vol. 34, pp. 22795–22807.
29. Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y.T.; Li, Y.; Lundberg, S.; et al. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712* **2023**.
30. Li, Y.; Ildiz, M.E.; Papailiopoulos, D.; Oymak, S. Transformers as Algorithms: Generalization and Stability in In-context Learning. *arXiv preprint arXiv:2301.07067* **2023**.
31. Hahn, M. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics* **2020**, *8*, 156–171.
32. Liu, B.; Ash, J.T.; Goel, S.; Krishnamurthy, A.; Zhang, C. Transformers learn shortcuts to automata. *arXiv preprint arXiv:2210.10749* **2022**.
33. Wei, J.; Wei, J.; Tay, Y.; Tran, D.; Webson, A.; Lu, Y.; Chen, X.; Liu, H.; Huang, D.; Zhou, D.; et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846* **2023**.
34. Wang, S.; Li, B.Z.; Khabsa, M.; Fang, H.; Ma, H. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768* **2020**.
35. Choromanski, K.M.; Likhosherstov, V.; Dohan, D.; Song, X.; Gane, A.; Sarlos, T.; Hawkins, P.; Davis, J.Q.; Mohiuddin, A.; Kaiser, L.; et al. Rethinking Attention with Performers. In Proceedings of the International Conference on Learning Representations, 2021.

 doi:10.20944/preprints202502.2271.v1

36. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The Long-Document Transformer. *arXiv:2004.05150* **2020**.
37. Qiu, J.; Ma, H.; Levy, O.; Yih, W.t.; Wang, S.; Tang, J. Blockwise Self-Attention for Long Document Understanding. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, 2020, pp. 2555–2565.
38. Alberti, S.; Dern, N.; Thesing, L.; Kutyniok, G. Sumformer: Universal Approximation for Efficient Transformers. In Proceedings of the Topological, Algebraic and Geometric Learning Workshops 2023. PMLR, 2023, pp. 72–86.
39. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to algorithms*; MIT press, 2022.
40. Weiss, G.; Goldberg, Y.; Yahav, E. Thinking like transformers. In Proceedings of the International Conference on Machine Learning. PMLR, 2021, pp. 11080–11090.
41. Peng, H.; Pappas, N.; Yogatama, D.; Schwartz, R.; Smith, N.A.; Kong, L. Random feature attention. *arXiv preprint arXiv:2103.02143* **2021**.
42. Loshchilov, I.; Hutter, F. Fixing weight decay regularization in adam **2017**.
43. Liu, B.; Ash, J.T.; Goel, S.; Krishnamurthy, A.; Zhang, C. Transformers Learn Shortcuts to Automata. In Proceedings of the The Eleventh International Conference on Learning Representations, 2023.
44. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. Language models are unsupervised multitask learners. *OpenAI blog* **2019**, *1*, 9.
45. Zniyed, Y.; Nguyen, T.P.; et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems* **2024**.
46. Wei, C.; Chen, Y.; Ma, T. Statistically meaningful approximation: a case study on approximating turing machines with transformers. *Advances in Neural Information Processing Systems* **2022**, *35*, 12071–12083.
47. Tarzanagh, D.A.; Li, Y.; Thrampoulidis, C.; Oymak, S. Transformers as Support Vector Machines. *arXiv preprint arXiv:2308.16898* **2023**.
48. Goel, S.; Kakade, S.M.; Kalai, A.T.; Zhang, C. Recurrent Convolutional Neural Networks Learn Succinct Learning Algorithms. In Proceedings of the Advances in Neural Information Processing Systems; Oh, A.H.; Agarwal, A.; Belgrave, D.; Cho, K., Eds., 2022.
49. Sun, Y.; Dong, L.; Huang, S.; Ma, S.; Xia, Y.; Xue, J.; Wang, J.; Wei, F. Retentive Network: A Successor to Transformer for Large Language Models (2023). *URL http://arxiv. org/abs/2307.08621 v1*.
50. OpenAI. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774* **2023**.
51. Von Oswald, J.; Niklasson, E.; Randazzo, E.; Sacramento, J.; Mordvintsev, A.; Zhmoginov, A.; Vladymyrov, M. Transformers learn in-context by gradient descent. In Proceedings of the International Conference on Machine Learning. PMLR, 2023, pp. 35151–35174.
52. Keles, F.D.; Wijewardena, P.M.; Hegde, C. On the computational complexity of self-attention. In Proceedings of the International Conference on Algorithmic Learning Theory. PMLR, 2023, pp. 597–619.
53. Yao, S.; Peng, B.; Papadimitriou, C.; Narasimhan, K. Self-Attention Networks Can Process Bounded Hierarchical Languages. In Proceedings of the Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 3770–3785.
54. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* **2016**.
55. Chan, S.C.; Santoro, A.; Lampinen, A.K.; Wang, J.X.; Singh, A.K.; Richemond, P.H.; McClelland, J.; Hill, F. Data Distributional Properties Drive Emergent In-Context Learning in Transformers. In Proceedings of the Advances in Neural Information Processing Systems, 2022.
56. Eldan, R.; Shamir, O. The power of depth for feedforward neural networks. In Proceedings of the Conference on learning theory. PMLR, 2016, pp. 907–940.
57. Feng, G.; Gu, Y.; Zhang, B.; Ye, H.; He, D.; Wang, L. Towards Revealing the Mystery behind Chain of Thought: a Theoretical Perspective. *Advances in Neural Information Processing Systems* **2023**.
58. 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* **2023**.
59. Prystawski, B.; Goodman, N.D. Why think step-by-step? Reasoning emerges from the locality of experience. *arXiv preprint arXiv:2304.03843* **2023**.
60. Vyas, A.; Katharopoulos, A.; Fleuret, F. Fast transformers with clustered attention. In Proceedings of the Advances in Neural Information Processing Systems, 2020, Vol. 33, pp. 21665–21674.

 doi:10.20944/preprints202502.2271.v1

61. 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* **2023**.
62. Merrill, W.; Sabharwal, A. The Parallelism Tradeoff: Limitations of Log-Precision Transformers. *Transactions of the Association for Computational Linguistics* **2023**.