Rethinking Transformers for Efficiency and Scalability

Huang Jiahao[1], Yawen Bao[1]

[1]Department of Computer Science, Tsinghua University, Beijing, China
firstname.lastname@tsinghua.edu.cn

Abstract. Machine learning has experienced a paradigm shift with the advent of Transformers, which have set new benchmarks across natural language processing, computer vision, speech processing, and biological sequence analysis. Despite their transformative impact, the quadratic complexity of the standard self-attention mechanism presents significant computational and memory challenges, especially for long sequences and resource-constrained settings. To overcome these limitations, efficient Transformer architectures have emerged, introducing novel approaches to reduce computational overhead while preserving model performance. This survey provides a detailed exploration of efficient Transformer methods, categorizing them into sparse attention mechanisms, low-rank approximations, memory-efficient architectures, and hybrid models. We analyze the trade-offs associated with these techniques and their implications for tasks such as long-form document understanding, high-resolution image processing, speech recognition, genomics, and multimodal learning. Additionally, we highlight key challenges, including the scalability to ultra-long sequences, robustness to diverse data distributions, and hardware-software optimization.
Looking forward, we discuss promising directions for future research, such as adaptive attention mechanisms, neural architecture search, integration with emerging hardware technologies, and sustainable AI practices. By addressing these challenges, efficient Transformers have the potential to further democratize access to advanced machine learning tools, making them more scalable, sustainable, and accessible for a wide range of applications.
Keywords: Machine learning, Transformers, efficient architectures, sparse attention, scalability, multimodal learning, sustainable AI.

1 Introduction

Transformers have emerged as a cornerstone of modern deep learning, revolutionizing fields such as natural language processing (NLP), computer vision, and speech processing [1]. Since the introduction of the Transformer architecture in the seminal work by Vaswani et al., its self-attention mechanism has set new benchmarks across a myriad of tasks, ranging from machine translation to image recognition [2]. Despite its remarkable success, the standard Transformer architecture comes with significant computational and memory overheads, particularly when applied to large-scale datasets or long sequences [3]. This computational burden is primarily attributed to the quadratic complexity of the self-attention mechanism, which scales with the sequence length [4]. As a result, scaling Transformers to longer sequences or resource-constrained environments has become a critical area of research. The burgeoning interest in efficient Transformers has been driven by both practical and theoretical considerations [5]. On the practical side, real-world applications often demand processing extensive data streams in real time, such as video streams, sensor data, or long textual documents [6]. These scenarios necessitate models that are not only accurate but also computationally efficient and scalable. Furthermore, the widespread deployment of Transformer models in edge devices, mobile platforms, and other resource-constrained settings underscores the need for lightweight and efficient variants. On the theoretical front, the exploration of sparse, low-rank, and structured approximations of attention mechanisms has shed light on fundamental trade-offs between expressiveness, efficiency, and interpretability [7]. Over the past few years, a plethora of methods have been proposed to mitigate the inefficiencies of the standard Transformer architecture [8]. These approaches can be broadly categorized into four main strategies: (1) reducing the complexity of the self-attention mechanism through sparsity or locality constraints, (2) leveraging low-rank approximations to model the attention matrix, (3) developing memory-efficient architectures by introducing recurrent or sliding-window mechanisms, and (4) employing hybrid models that combine the strengths of Transformers with convolutional or recurrent neural networks. While these innovations have significantly advanced the state of efficient Transformers, the landscape remains fragmented, with diverse methodologies, benchmarks, and evaluation protocols. The objective of this survey is to provide a comprehensive overview of the advancements in efficient Transformer architectures [9]. We aim to bridge the gap between theory and practice by systematically categorizing and analyzing existing methods, highlighting their strengths, limitations, and application domains [10]. Unlike previous reviews, which often focus on a narrow subset of methods or applications, this survey adopts a holistic approach, encompassing the latest developments across various modalities and tasks [11]. In this survey, we address several key questions: What are the fundamental limitations of the standard Transformer architecture, and how do they manifest in different application scenarios [12]? How have researchers approached the problem of designing efficient Transformers, and what are the theoretical underpinnings of these methods [13]? What are the trade-offs between computational efficiency, model expressiveness, and empirical performance [14]? Finally, what are the open challenges and future directions in this rapidly evolving field [15]? The structure of this survey is as follows [16]. In Section 2, we provide a brief overview of the Transformer architecture, emphasizing its computational challenges [17]. Section 3 delves into the diverse methodologies proposed to enhance the efficiency of Transformers, categorizing them into major themes and approaches. Section 4 explores the application domains where efficient Transformers have made significant impacts, from NLP to computer vision. Finally, Section 5 outlines the open challenges and promising directions for future research [18]. In summary, this survey aims to serve as a definitive reference for researchers, practitioners, and enthusiasts interested in the field of efficient Transformers [19]. By providing a detailed and structured examination of the existing literature, we hope to inspire further innovations and foster a deeper understanding of this exciting and impactful area of research [20].

2 Background

The Transformer architecture, introduced by Vaswani et al., has become a fundamental building block in modern deep learning [21]. Its core innovation, the self-attention mechanism, allows the model to capture dependencies across sequences without the need for recurrent or convolutional operations. This section provides a detailed overview of the standard Transformer architecture, its components, and the computational challenges that motivate the development of efficient variants [22].

2.1 The Transformer Architecture

The Transformer consists of an encoder-decoder structure, though many applications, such as BERT and GPT, use only the encoder or decoder [23]. Each component is built from a stack of layers, where each layer comprises two primary subcomponents: multi-head self-attention and feedforward neural networks [24].

Multi-Head Self-Attention The self-attention mechanism computes a weighted sum of all input tokens, enabling the model to focus on relevant parts of the sequence [25]. Given a sequence of input tokens $\mathbf{X} \in \mathbb{R}^{n \times d}$, self-attention involves three key steps:

- **Linear Transformations:** The input $\mathbf{X}$ is linearly transformed into query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$) matrices using learned projection matrices $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ [26].
- **Attention Computation:** The attention scores are calculated as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \tag{1}$$

  where d_k is the dimensionality of the key vectors, and the softmax ensures that the attention weights sum to one.
- **Multi-Head Mechanism:** Multiple attention heads are computed in parallel, allowing the model to attend to different parts of the sequence [27]. The outputs of these heads are concatenated and linearly projected [28].

Feedforward Layers Each layer of the Transformer includes a position-wise feedforward neural network (FFN) applied independently to each token [29]. This component enhances the model's representational power by learning complex, non-linear transformations [30].

Positional Encodings Since the Transformer lacks inherent sequential inductive bias, positional encodings are added to the input embeddings to provide information about the order of tokens [31]. These encodings can be either fixed (e.g., sinusoidal) or learned [32].

2.2 Computational Challenges

While the Transformer architecture has demonstrated unparalleled performance, its design incurs significant computational and memory costs, especially when processing long sequences [33]. The primary bottleneck lies in the self-attention mechanism, which has a complexity of $O(n^2 \cdot d)$, where n is the sequence length and d is the embedding dimension [34]. This quadratic scaling arises from the computation of pairwise attention scores between all tokens. The challenges become particularly pronounced in the following scenarios:

- **Long Sequences:** Tasks involving long input sequences, such as document modeling, video analysis, or protein folding, exacerbate the quadratic complexity, leading to prohibitive memory and compute requirements [35].
- **Large-Scale Models:** Modern applications often involve billions of parameters, further amplifying the computational demands [36].
- **Resource-Constrained Environments:** Deploying Transformers on edge devices or low-resource hardware necessitates lightweight and efficient architectures [19].

2.3 Motivation for Efficient Transformers

The limitations of the standard Transformer have spurred extensive research into efficient alternatives. By reducing the complexity of self-attention, optimizing memory usage, and introducing sparsity, researchers aim to make Transformers more scalable and adaptable to diverse tasks and hardware constraints [37]. The subsequent sections of this survey delve into these efforts, categorizing and analyzing the myriad approaches proposed in the literature.

3 Methods for Efficient Transformers

Numerous approaches have been proposed to enhance the efficiency of Transformer architectures, addressing the computational and memory challenges inherent in the standard design. These methods can be broadly categorized into four main strategies: sparse attention mechanisms, low-rank approximations, memory-efficient architectures, and hybrid models [38]. This section provides a detailed examination of these strategies, highlighting their core ideas, advantages, and limitations [39].

3.1 Sparse Attention Mechanisms

Sparse attention mechanisms aim to reduce the quadratic complexity of self-attention by restricting the number of token pairs for which attention is computed [40]. This sparsity is achieved through predefined patterns, learnable structures, or a combination of both [19].

Predefined Sparse Patterns Predefined sparse patterns impose fixed constraints on the attention computation. Examples include:

- **Local Attention:** Tokens attend only to their neighbors within a fixed window, reducing complexity to $O(n \cdot w)$, where w is the window size [41].
- **Global and Strided Patterns:** Selected tokens (e.g., special tokens or periodic intervals) attend globally, while others follow a strided pattern, balancing sparsity and coverage [42].
- **Block Sparse Attention:** The input sequence is divided into blocks, and attention is computed only within or between specific blocks.

Learnable Sparse Patterns Learnable patterns adapt the sparsity dynamically based on input data [43]. Notable methods include:

- **Routing-Based Attention:** Tokens are grouped into clusters, and attention is computed within and across clusters [44].
- **Content-Based Sparsity:** Attention weights are computed sparsely based on the similarity between tokens, retaining only the most relevant connections [45].

3.2 Low-Rank Approximations

Low-rank approximation techniques leverage the redundancy in the attention matrix to reduce its computational complexity [46]. These methods approximate the attention matrix using fewer parameters, achieving significant memory and time savings [47].

Kernel-Based Approximations Kernel-based methods replace the dot-product attention with kernel functions that decompose the attention matrix into low-rank components [48]. Notable examples include:

- **Linear Transformers:** Linearizing attention by approximating softmax with kernel functions reduces complexity to $O(n \cdot d)$ [49].
- **Performer:** Employing positive orthogonal random features for unbiased approximation of attention [50].

Projection-Based Approximations Projection-based methods decompose the input sequence into a lower-dimensional subspace, enabling efficient computation [51]. Examples include:

– **Singular Value Decomposition (SVD):** Approximating the attention matrix using its dominant singular vectors [52].
– **Attention via Clustering:** Grouping tokens and computing attention within cluster representatives [53].

3.3 Memory-Efficient Architectures

Memory-efficient designs aim to reduce storage requirements by reusing or compressing intermediate representations during computation [54].

Recurrent Architectures Recurrent models process sequences incrementally, maintaining a fixed-size memory state. This approach is particularly suitable for streaming applications.

Sliding Window and Chunking Models such as the Longformer and BigBird adopt sliding window or chunking strategies to limit attention computation to smaller sequence segments.

3.4 Hybrid Models

Hybrid models combine the Transformer architecture with other neural network paradigms, such as convolutional or recurrent layers, to exploit their complementary strengths [55].

Transformer-ConvNet Hybrids Incorporating convolutional layers enhances local feature extraction while retaining the global modeling capabilities of attention [56].

Transformer-RNN Hybrids Recurrent layers are integrated to capture sequential dependencies efficiently, especially in streaming and online settings.

3.5 Comparison of Methods

Table 1 provides a comparative summary of the methods discussed, highlighting their complexity, advantages, and typical application domains [57].

Table 1. Comparison of Efficient Transformer Methods

Method	Complexity	Advantages	Applications
Sparse Attention	$O(n \cdot w)$	Scalable to long sequences	NLP, Vision
Low-Rank Approximations	$O(n \cdot d)$	Memory-efficient	Large-scale models
Memory-Efficient Architectures	Varies	Real-time applications	Streaming data
Hybrid Models	Varies	Flexibility, modularity	Multi-modal tasks

4 Applications of Efficient Transformers

The versatility and adaptability of efficient Transformers have enabled their deployment across a wide range of applications [58]. By addressing the computational limitations of the standard Transformer, these architectures have unlocked new possibilities in tasks that require processing long sequences, operating in real-time, or functioning within resource-constrained environments [59]. This section provides an overview of the primary application domains, highlighting the role of efficient Transformers in each.

4.1 Natural Language Processing

Natural Language Processing (NLP) remains one of the most prominent domains for Transformer models. Efficient Transformers have been particularly impactful in the following NLP tasks:

Document Understanding Long-form document understanding tasks, such as summarization, question answering, and legal or scientific text analysis, often involve sequences that exceed the token limits of standard Transformers [60]. Efficient architectures like Longformer and BigBird excel in these scenarios by enabling attention over long sequences without incurring quadratic complexity [61].

Dialogue Systems and Conversational AI Efficient Transformers facilitate real-time, multi-turn conversations in dialogue systems by reducing latency and memory overhead [12]. Models such as Reformer enable large-scale deployments of conversational agents on low-resource devices.

4.2 Computer Vision

The introduction of Vision Transformers (ViTs) [62] marked a paradigm shift in computer vision, demonstrating that attention mechanisms can outperform convolutional neural networks (CNNs) on various image recognition tasks. Efficient Transformers extend these capabilities to more demanding applications:

High-Resolution Image Processing Tasks such as medical imaging, satellite image analysis, and video frame processing benefit from sparse and hierarchical attention mechanisms, which reduce memory and computation requirements [63].

Video Understanding Video understanding tasks, including action recognition and video summarization, involve analyzing sequences of frames [64]. Efficient architectures such as TimeSformer reduce computational overhead by leveraging sparse temporal attention [65].

4.3 Speech Processing

Efficient Transformers have also made strides in speech processing, where sequence lengths can be particularly long due to high sampling rates [66]. Applications include:

Speech Recognition By integrating sparse or low-rank attention mechanisms, efficient Transformers improve the scalability of automatic speech recognition systems while maintaining high accuracy [67].

Speech Synthesis Speech synthesis models, such as text-to-speech systems, benefit from efficient attention mechanisms to handle long-duration audio sequences in a memory-efficient manner.

4.4 Biological Sequence Analysis

The analysis of biological sequences, such as DNA, RNA, and proteins, often involves extremely long sequences, making efficient Transformers a natural fit [68]. Applications in this domain include:

Genomics Efficient Transformers are employed for tasks such as variant calling, genome annotation, and sequence alignment, where scalability and accuracy are paramount.

Protein Structure Prediction In protein structure prediction, long-range dependencies in amino acid sequences are critical. Efficient architectures enable the processing of these sequences at scale, as demonstrated by AlphaFold's success [69].

4.5 Multimodal Applications

Efficient Transformers have also shown promise in multimodal tasks, where information from different modalities (e.g., text, images, audio) must be integrated:

Visual Question Answering By combining efficient attention mechanisms with modality-specific encoders, efficient Transformers enable real-time inference for tasks like visual question answering and image captioning [32].

Robotics and Autonomous Systems Efficient Transformers are used in robotics for tasks such as sensor fusion and decision-making in real-time environments, leveraging their ability to handle diverse input modalities [70].

4.6 Edge Computing and Resource-Constrained Settings

One of the most impactful applications of efficient Transformers is their deployment in edge computing scenarios. Tasks such as real-time translation, object detection on mobile devices, and IoT sensor analysis require lightweight models that can operate with limited computational resources [71]. Memory-efficient architectures and sparse attention mechanisms have enabled significant progress in this area [72].

4.7 Summary of Applications

Efficient Transformers have expanded the scope of attention-based models, making them feasible for tasks that were previously constrained by computational and memory limitations [73]. Table 2 summarizes the key application domains and the corresponding efficient Transformer methods commonly used [74].

Table 2. Summary of Application Domains for Efficient Transformers

Domain	**Key Tasks**	**Efficient Transformer Methods**
NLP	Document understanding, dialogue systems	Sparse attention, low-rank approximations
Computer Vision	High-resolution image processing, video understanding	Hybrid models, hierarchical attention
Speech Processing	Speech recognition, synthesis	Memory-efficient architectures
Biological Analysis	Genomics, protein structure prediction	Sparse attention, kernel approximations
Multimodal Tasks	Visual question answering, robotics	Hybrid models, cross-modal attention
Edge Computing	Real-time translation, IoT analysis	Lightweight models, memory optimization

5 Challenges and Future Directions

While efficient Transformers have made significant progress in addressing the limitations of standard architectures, several challenges remain [75]. This section outlines key obstacles and highlights promising directions for future research.

5.1 Challenges

Balancing Efficiency and Expressiveness Many efficient Transformer methods achieve computational savings by introducing approximations, such as sparsity or low-rank representations [76]. However, these approximations can degrade model performance on tasks requiring fine-grained attention or long-range dependencies. Striking the right balance between efficiency and expressiveness remains a critical challenge [77].

Scalability to Ultra-Long Sequences While many efficient Transformers handle moderately long sequences (e.g., several thousand tokens) effectively, scaling to ultra-long sequences, such as entire books or genome sequences, remains a challenge [78]. Existing methods often trade off memory for computational savings, which may not always be feasible for extremely large datasets [79].

Robustness and Generalization Efficient Transformers often rely on assumptions about sparsity or structure in the data [80]. These assumptions may not hold in all scenarios, leading to brittleness and reduced generalization to out-of-domain tasks. Developing more robust architectures that adapt dynamically to diverse input distributions is an ongoing challenge [81].

Hardware and Software Optimization Efficient Transformer methods often require specialized hardware or software optimizations to fully realize their potential [82]. For example, sparse attention patterns or kernel-based approximations may not be efficiently implemented on all hardware platforms [83]. Bridging the gap between algorithmic innovations and practical deployment remains a key hurdle.

5.2 Future Directions

Dynamic and Adaptive Attention One promising direction is the development of dynamic and adaptive attention mechanisms that can adjust their computational budget based on input complexity or task requirements [84]. Such mechanisms could enable more efficient resource utilization while maintaining performance [85].

Neural Architecture Search (NAS) for Efficiency Neural architecture search has shown promise in automating the design of deep learning models [86]. Applying NAS to discover optimal configurations for efficient Transformers, tailored to specific tasks and hardware constraints, is an exciting avenue for future work [87].

Integration with Emerging Hardware Emerging hardware technologies, such as neuromorphic computing, photonic accelerators, and quantum processors, offer opportunities to rethink the design of efficient Transformers [88]. Collaborations between model designers and hardware engineers could lead to architectures that exploit these technologies to their fullest potential [89].

Multimodal and Cross-Task Efficiency Efficient Transformers have primarily focused on improving performance within individual domains [62]. Future research could explore architectures that generalize across modalities and tasks, enabling seamless integration of text, vision, and audio data while maintaining efficiency [90].

Theoretical Understanding of Efficiency The design of efficient Transformers has largely been empirical, with limited theoretical analysis of why certain methods work better than others [91]. A deeper theoretical understanding of the trade-offs between sparsity, low-rank approximations, and model expressiveness could guide the development of new architectures.

Sustainability and Green AI Efficient Transformers play a critical role in reducing the environmental impact of deep learning. Future work could focus on developing architectures that prioritize energy efficiency and sustainability, making large-scale models accessible to a broader range of researchers and practitioners [92].

5.3 Conclusion

Efficient Transformers have made remarkable strides in addressing the scalability challenges of the standard Transformer architecture [93]. However, the field is still in its infancy, with numerous opportunities for innovation. By addressing the challenges outlined above and pursuing the proposed future directions, researchers can further advance the state of the art, enabling efficient and scalable Transformers for a wider range of applications [94].

6 Conclusion

Transformers have emerged as a transformative architecture across a wide range of domains, but their high computational and memory requirements have posed significant challenges to their scalability and deployment. Efficient Transformers have addressed these limitations through innovative methods, including sparse attention mechanisms, low-rank approximations, memory-efficient architectures, and hybrid models. These advancements have enabled Transformers to process longer sequences, operate in real-time, and function on resource-constrained devices, opening up new possibilities in natural language processing, computer vision, speech processing, biological sequence analysis, and beyond.

This survey has provided a comprehensive overview of the methods, applications, and challenges associated with efficient Transformers. By categorizing and analyzing the core strategies, we have highlighted the trade-offs between computational efficiency and model expressiveness. Sparse attention mechanisms, for instance, offer significant speedups but may struggle with tasks requiring fine-grained contextual understanding. Similarly, low-rank approximations and memory-efficient designs provide scalability benefits but often rely on assumptions that may limit their robustness.

Despite these challenges, the progress in this field has been remarkable, and the potential for further innovation is vast. Future research directions, such as dynamic attention mechanisms, integration with emerging hardware, and the exploration of multimodal efficiency, promise to push the boundaries of what efficient Transformers can achieve. Furthermore, the growing emphasis on sustainability and green AI underscores the importance of designing architectures that not only excel in performance but also minimize environmental impact.

Efficient Transformers represent a critical step toward democratizing the power of attention-based models, making them accessible to a broader audience and applicable to a wider range of problems. As the field continues to evolve, collaboration between researchers, practitioners, and hardware developers will be essential to realize the full potential of these architectures. By addressing the challenges and embracing the opportunities outlined in this survey, we can look forward to a future where Transformers are not only powerful but also efficient, scalable, and sustainable.

References

1. Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Proceeding of NeurIPS*, 2014.
2. Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.
3. Shengjie Luo, Shanda Li, Tianle Cai, Di He, Dinglan Peng, Shuxin Zheng, Guolin Ke, Liwei Wang, and Tie-Yan Liu. Stable, fast and accurate: Kernelized attention with relative positional encoding. In *Advances in Neural Information Processing Systems*, volume 34, pages 22795–22807, 2021.
4. Zihang Dai, Guokun Lai, Yiming Yang, and Quoc V Le. Funnel-transformer: Filtering out sequential redundancy for efficient language processing. *Proceedings of NeurIPS*, 2020.
5. Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
6. Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*, 2019.
7. Josh Alman and Zhao Song. Fast attention requires bounded entries. *arXiv preprint arXiv:2302.13214*, 2023.
8. Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.
9. Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021.
10. Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics.
11. Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.
12. Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Proceedings of TACL*, 2020.

13. Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*, 2019.
14. Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023.
15. Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, July 2017. Association for Computational Linguistics.
16. Yi Tay, Vinh Q Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. Charformer: Fast character transformers via gradient-based subword tokenization. *arXiv preprint arXiv:2106.12672*, 2021.
17. Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. *Proceedings of ICLR*, 2021.
18. Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.
19. Yassine Zniyed, Thanh Phuong Nguyen, et al. Efficient tensor decomposition-based filter pruning. *Neural Networks*, 178:106393, 2024.
20. Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *Proceedings of EMNLP*, 2020.
21. Xuezhe Ma, Xiang Kong, Sinong Wang, Chunting Zhou, Jonathan May, Hao Ma, and Luke Zettlemoyer. Luna: Linear unified nested attention. In *Proceedings of NeurIPS 2021*, 2021.
22. Gonçalo M Correia, Vlad Niculae, and André FT Martins. Adaptively sparse transformers. *Proceedings of EMNLP*, 2019.
23. Hanrui Wang, Zhanghao Wu, Zhijian Liu, Han Cai, Ligeng Zhu, Chuang Gan, and Song Han. Hat: Hardware-aware transformers for efficient natural language processing. *arXiv preprint arXiv:2005.14187*, 2020.
24. Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nystr\" omformer: A nystr\" om-based algorithm for approximating self-attention. *Proceedings of AAAI*, 2021.
25. Michael Hahn and Navin Goyal. A theory of emergent in-context learning as implicit structure induction. *arXiv preprint arXiv:2303.07971*, 2023.
26. Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with gumbel-sinkhorn networks. *arXiv preprint arXiv:1802.08665*, 2018.
27. Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. *Proceedings of ICLR 2019*, 2018.
28. Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020.
29. Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural

image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.
30. Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Q-bert: Hessian based ultra low precision quantization of bert. 2020.
31. Yi Tay, Mostafa Dehghani, Jai Gupta, Dara Bahri, Vamsi Aribandi, Zhen Qin, and Donald Metzler. Are pre-trained convolutions better than pre-trained transformers? *arXiv preprint arXiv:2105.03322*, 2021.
32. Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.
33. Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
34. Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.
35. Clayton Sanford, Daniel Hsu, and Matus Telgarsky. Representational strengths and limitations of transformers. *arXiv preprint arXiv:2306.02896*, 2023.
36. Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pages 3744–3753, 2019.
37. Angela Fan, Pierre Stock, Benjamin Graham, Edouard Grave, Remi Gribonval, Herve Jegou, and Armand Joulin. Training with quantization noise for extreme fixed-point compression. *arXiv preprint arXiv:2004.07320*, 2020.
38. Richard Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–515, 1954.
39. Jiezhong Qiu, Hao Ma, Omer Levy, Scott Wen tau Yih, Sinong Wang, and Jie Tang. Blockwise self-attention for long document understanding, 2020.
40. Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.
41. Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
42. Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
43. Apoorv Vyas, Angelos Katharopoulos, and François Fleuret. Fast transformers with clustered attention. *Proceedings of NeurIPS*, 2020.
44. Jiezhong Qiu, Hao Ma, Omer Levy, Scott Wen-tau Yih, Sinong Wang, and Jie Tang. Blockwise self-attention for long document understanding. *arXiv preprint arXiv:1911.02972*, 2019.

45. Beidi Chen, Tri Dao, Eric Winsor, Zhao Song, Atri Rudra, and Christopher Ré. Scatterbrain: Unifying sparse and low-rank attention. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.
46. Wenhan Xiong, Barlas Oğuz, Anchit Gupta, Xilun Chen, Diana Liskovich, Omer Levy, Wen-tau Yih, and Yashar Mehdad. Simple local attentions remain competitive for long-context tasks. *arXiv preprint arXiv:2112.07210*, 2021.
47. Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
48. Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
49. Jorge Pérez, Pablo Barceló, and Javier Marinkovic. Attention is turing complete. *The Journal of Machine Learning Research*, 22(1):3463–3497, 2021.
50. François Lagunas, Ella Charlaix, Victor Sanh, and Alexander M Rush. Block pruning for faster transformers. *Proceedings of EMNLP 2021*, 2021.
51. Joshua Ainslie, Santiago Ontanon, Chris Alberti, Philip Pham, Anirudh Ravula, and Sumit Sanghai. Etc: Encoding long and structured data in transformers. *Proceedings of EMNLP*, 2020.
52. Hanxiao Liu, Zihang Dai, David R So, and Quoc V Le. Pay attention to mlps. *Proceedings of NeurIPS*, 2021.
53. Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathy Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. Glam: Efficient scaling of language models with mixture-of-experts, 2021.
54. Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. Sparse sinkhorn attention. In *International Conference on Machine Learning*, pages 9438–9447. PMLR, 2020.
55. Bingbin Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. *arXiv preprint arXiv:2210.10749*, 2022.
56. Niki Parmar, Prajit Ramachandran, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. In *Advances in Neural Information Processing Systems*, pages 68–80, 2019.
57. Markus N Rabe and Charles Staats. Self-attention does not need o (nˆ 2) memory. *arXiv preprint arXiv:2112.05682*, 2021.
58. Lukasz Kaiser, Aidan N Gomez, and Francois Chollet. Depthwise separable convolutions for neural machine translation. *Proceedings of ICLR*, 2017.
59. Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *Proceedings of NAACL 2018*, 2017.
60. Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.
61. Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark

for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.
62. Yingcong Li, M Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and stability in in-context learning. *arXiv preprint arXiv:2301.07067*, 2023.
63. Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A Smith, and Lingpeng Kong. Random feature attention. *arXiv preprint arXiv:2103.02143*, 2021.
64. Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In *Advances in Neural Information Processing Systems*, 2022.
65. Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. On the ability and limitations of transformers to recognize formal languages. *arXiv preprint arXiv:2009.11264*, 2020.
66. Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019.
67. Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*, 2020.
68. Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*, 2023.
69. Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2017.
70. Davoud Ataee Tarzanagh, Yingcong Li, Christos Thrampoulidis, and Samet Oymak. Transformers as support vector machines. *arXiv preprint arXiv:2308.16898*, 2023.
71. Yiding Hao, Dana Angluin, and Robert Frank. Formal language recognition by hard attention transformers: Perspectives from circuit complexity. *Transactions of the Association for Computational Linguistics*, 10:800–810, 2022.
72. Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *Proceedings of EMNLP*, 2015.
73. Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *Proceedings of ICLR*, 2016.
74. Yi Tay, Aston Zhang, Luu Anh Tuan, Jinfeng Rao, Shuai Zhang, Shuohang Wang, Jie Fu, and Siu Cheung Hui. Lightweight and efficient neural natural language processing with quaternion networks. *Proceedings of ACL*, 2019.
75. Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.
76. Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.
77. Anonymous. Scaling laws vs model architectures: How does inductive bias influence scaling? an extensive empirical study on language tasks. *ACL Rolling Review, September*, 2021.
78. Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *Proceedings of ICLR 2016*, 2015.
79. Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. Adaptive attention span in transformers. *arXiv preprint arXiv:1905.07799*, 2019.

80. Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*, pages 1614–1623, 2016.
81. Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy, July 2019. Association for Computational Linguistics.
82. Stephanie C.Y. Chan, Adam Santoro, Andrew Kyle Lampinen, Jane X Wang, Aaditya K Singh, Pierre Harvey Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. In *Advances in Neural Information Processing Systems*, 2022.
83. Yassine Zniyed, Thanh Phuong Nguyen, et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.
84. Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019.
85. Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.
86. Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *Proceedings of ICLR*, 2017.
87. Sebastian Jaszczur, Aakanksha Chowdhery, Afroz Mohiuddin, Łukasz Kaiser, Wojciech Gajewski, Henryk Michalewski, and Jonni Kanerva. Sparse is enough in scaling transformers. *Advances in Neural Information Processing Systems*, 34, 2021.
88. Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. Designing effective sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.
89. Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. *arXiv preprint arXiv:2006.16236*, 2020.
90. Shunyu Yao, Binghui Peng, Christos Papadimitriou, and Karthik Narasimhan. Self-attention networks can process bounded hierarchical languages. In *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)*, pages 3770–3785, 2021.
91. OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
92. William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*, 2021.
93. Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Proceedings of NeurIPS*, 2019.
94. Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964.