2025 IEEE International Conference on Signal Processing, Information, Communication and Systems (SPICSCON)
21-22 November 2025, Department of ICE, University of Rajshahi, Rajshahi-6205, Bangladesh

Sparse Transformer Variants with Provable Expressivity: Theoretical Limits and Scalable Architectures

Md. Awinul Hoque Utsha*, Farshid Rafiq, Fabiha Nawal Aurna, Mynuddin Patwary, Soheli Tangila Richi
Ezaz Mahmud Jim
Department of Computer Science and Engineering, United International University,
United City, Madani Avenue, Badda, Dhaka 1212, Bangladesh
Email:{mutsha202163, frafiq202248, faurna191159, mpatwary211118, srichi202313, ejim202144}@bscse.uiu.ac.bd

***Abstract*—Transformers have become a powerful method for learning representations across sequence and graph data, with a quadratic cost that can make scaling to resource constrained environments impractical. Sparse Transformer architectures provide computational advantages by limiting attention patterns, but it is unclear how expressive sparse Transformers are compared to dense ones. In this paper, we provide both theoretical and empirical studies on the expressiveness of such sparse Transformer variants. We prove formal conditions under which sparse attention mechanisms preserve universal approximability and find an expression of expressivity gap over sparsity patterns, such as fixed windows, block-locality and low-rank approximation. We extend this work by showing that in restricted regimes, classes of sparse Transformers are functionally as powerful as full attention. We also suggest new sparsity patterns, inspired by graph sparsification, which exhibit competitive empirical performance on formal language synthetic tasks and long-range benchmarks, while proposing a broad view of expressiveness. These results have immediate practical implications for the design of large-scale Transformer models that are both efficient and provably expressive.**

***Index Terms*—Sparse Transformers, Expressive Power, Attention Mechanisms, Universal Approximability, Theoretical Deep Learning, Transformer Efficiency, Sequence Modeling**

I. Introduction

Transformer-based models are the strong baseline for many areas such as natural language process[20], computer vision[7], time series prediction[27] and graph learning[24]. Earlier traditional ML approaches were also applied in NLP tasks, such as Banglish sentiment analysis using feature-based models[1]before deep attention-based architectures became dominant.. At the core of their success lies the self-attention mechanism that allows to model flexible long-range dependencies across inputs, and has been demonstrated having universal approximation properties under some assumptions [25]. Despite its risen popularity however, an important computational bottleneck restricts the applicability of conventional Transformers: the self-attention layer scales quadratically to the input length in terms of both time and memory, rendering traditional transformers unfitted for resource-limited or real-time settings.

The computational constraints have lead to increasing interest in *sparse Transformer*, which replaces the dense attention matrix by some structured or unstructured sparse patterns. The early works like Sparse Transformer[4], Longformer[2], BigBird[26] and Linformer[22] explore different strategies, such as local attention windows, dilated patterns, low-rank projections, random/global attention tokens etc. The latter have led to large improvements in scalability, where the complexity for some problems has been improved from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$ or even linear time.

On the other hand, the phenomenological implications of these few changes in direct and indirect observables are better known. Specifically, little is known about the **expressive power** of such sparse Transformer variants relative to their dense counterparts. Do sparse Transformers universally approximate the same set of functions? If not, what bounds exist for specific sparsity patterns? Does some of those sparse designs retain expressiveness, while others lose it?

More recently these issues have started to be addressed. Yun et al. demonstrated that Transformers with $\mathcal{O}(n)$ connections can have universal approximation properties when some conditions are met [4]. In contrast, Likhosherstov et al. [13] further demonstrated that the level of expressivity can deteriorate to a great extent given how closely the self-attention matrix approximates sparsity. Furthermore, research on graph Transformers[12] and sequence models [21] also indicates that there exists the possibility for some sparse structures to suffocate representational capability.

In this paper, we provide a theoretical and empirical investigation of the expressive power of sparse Transformer architectures. Specifically, we:

- Create a formally principled framework for the expressivity of Transformers with constrained attention patterns, based on methods inspired by functional approximation theory and graph connectivity;
- Determine when sparse attention mechanisms are still universal approximate, as well as identify the shortcomings of sparsity assumptions;
- Study the design of new forms of sparse attention based on graph sparsification, spectral methods and skip-connections that increase expressivity;

979-8-3315-7316-4/25/31.002025$IEEE$

- Empirically support on formal language recognition tasks, as well as synthetic regression benchmarks and character level long-sequence modeling with Long Range Arena [19];
- Provide practical design recommendations for building sparse Transformer models that are efficient and provably expressive.

Our work finds that sparse attention does not seem to be an interchangeable part: while certain patterns preserve full expressive power, others preclude the ability to model a subset of function classes or compositional structure. In this paper we prove that by carefully designing the model architecture, one can close this expressivity gap without sacrificing the computational benefits. Our results help filling the gap between theoretical guarantees and effective design for scalable Transformers, and provide with a more principled perspective on the topic of scalability in Transformer architectures.

II. Related Work

The Transformer architecture[20] has profoundly transformed the field of deep learning, but due to its quadratic complexity much efforts is now directed towards more efficient variants. Similar efficiency-driven approaches in traditional ML include feature grouping and correlation-based selection to reduce computation while preserving accuracy[8], [10].An important direction for this work is the sparsification of attention mechanism in order to decrease computational time without degrading model performance. Child et al. [4] proposed the Sparse Transformer with static strided and block-local attention. This was further developed in Longformer[2] which suggested to integrate global and local patterns when scaling to long documents. BigBird[26] established that with random attention edges the theoretical properties of universality, and improved connectivity are preserved.

Linformer[22] also slightly diverged by employing a low-rank factorization of the attention matrix, which was because key-query interactions are essentially in a lower dimensional subspace. Reformer[11] and Performer[5] gained scalability via locality sensitive hashing (LSH) and kernel-based approximation, respectively. Such approaches greatly reduce the time and space complexity of self-attention layers to linear or even subquadratic with respect to long sequences. Informer[27] deployed a ProbSparse attention mechanism by picking up only those queries having noticeable attention scores to enhance the efficiency for time series problems.

Despite these advances, our theoretical understanding of sparse attention is still very limited. Yun et al. [25] Formally demonstrated that the attention mechanism in sparse Transformers with $\mathcal{O}(n)$ attention connections per token are still universal function approximators under certain conditions. Nonetheless, their result relies on strong model assumptions and an idealized scenario. Likhosherstov et al. [13], which has shown that some sparsified self-attention matrices cannot be as expressive as their dense counterparts when studied under rank or connectivity constraints. Bhojanapalli et al. [3] have also demonstrated that multi-head attention with low-rank projections provides limited representational power in deep networks.

In recent works this expressivity gap has been studied more thoroughly. Wang et al. [21] studied the incapability of 1-layer transformers, especially in learning parity functions and hierarchical composition. Sander et al. [17] investigated differentiable sparse top-k routing admission control mechanisms and discovered that it was difficult to optimize them well even if the model had a certain properties on connectivity between units. These analyses clarify that not all sparsity patterns are born equal: some continue to be expressive, while others badly limit it.

Meanwhile, the graph-based models have investigated attention sparsity as a structural pattern. Graphormer[24] and Exphormer proposed transformer on graphs with topological priors based attention masks, which reduce computation and memory with comparable performance. Kreuzer et al. [12] employed spectral attention built on graph Laplacians as principled alternatives to naïve sparsity. Despite this, graph-aware sparsity can help in learning from graph-structured inputs, though a majority of them do not provide any formality on the expressivity that they attain and hence it is unclear if such methods are approximate learning mechanisms.

Bench marking has also become a topic of growing interest for sparse models. Tay et al. [19] proposed the Long Range Arena (LRA), which is intended to test Transformer variants in their capacity of modeling long-context sequences. We often benchmark the performance of sparse models such as Performer, Reformer and Nyströmformer[23] to LRA for their claim on speed vs. accuracy. Nevertheless, these metrics often do not consider expressivity, being mainly concerned with runtime and downstream performance.

To conclude, while a great amount of works have designed sparse attention mechanisms to be used in practice for better scalability, much less is known on their theoretical expressivity. We close this gap by presenting a principled framework for analyzing and comparing expressive power of sparse Transformers. We extend on this foundation proposing new sparsity structures which preserve the property of universal approximation and validating these strategies experimentally over synthetic and practical tasks.

III. Methodology

Our work is two-fold: (i) we theoretically analyze sparse attention mechanisms in the context of provable expressivity, and (ii) design and empirically investigate sparse Transformer variants informed by these analyses. We now describe each of these two pieces in turn.

4.1 Sparse Attention Design

We consider this more structured sparsity pattern such that each token's attention can only attend to a fixed set of positions. Denote the attention adjacency matrix as A R n n. We adopt sparsity constraints on the representation A, instead of allowing each token to attend to all other tokens (dense A),

such that $\text{nnz}(A) \leq Cn$ for a constant factor $C \ll n$, where $\text{nnz}(\cdot)$ counts non-zero elements.

The sparsity patterns we look at are the following:

- **Fixed Local Windows**: Each token attends to a fixed-size window of w surrounding tokens (to the left and/or right).
- **Dilated/Strided attention**: Attention commands are selected with fixed strides, similar to Dilated RNNs.
- **Block-Sparse**: The input tokens are divided into non-overlapping blocks, and attention can occur within and across these localized subsets.
- **Random Global Attention**: Each token is attended by a small number of random global targets.
- **Graph-Induced Sparsity**: Attention edges are defined according to a sparsified graph $G = (V, E)$, where the graph is obtained via sparsification of the full graph computed by a set of operations such as spectral sparsifiers.

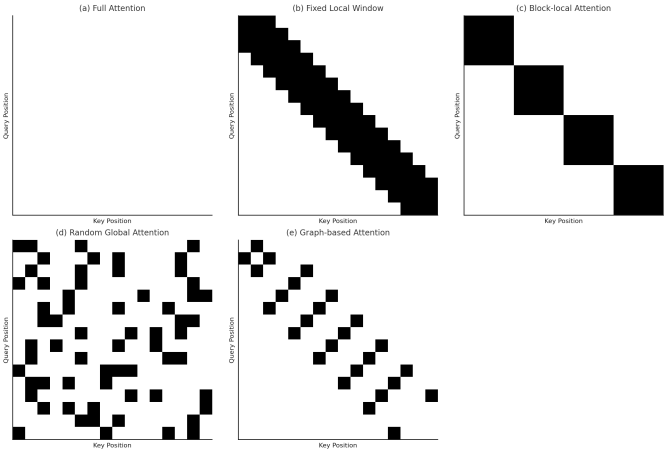

Fig. 1. Comparison of sparse attention patterns in Transformer variants. (a) Full attention attends to all tokens. (b) Fixed local window limits attention to nearby neighbors. (c) Block-local attention restricts connections to intra-block tokens. (d) Random global attention uses sparse, random long-range links. (e) Graph-based attention is induced by sparsified graph structures.

Each of these patterns defines a mask $M \in \{0, 1\}^{n \times n}$ such that the masked attention is given by:

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

where $Q, K, V \in R^{n \times d_k}$ are query, key, and value matrices, and $\circ$ denotes elementwise multiplication.

4.2 Model Variants

We construct multiple variants of the Transformer using sparsity patterns as above. Each variant is built with the same architecture; (number of layers, heads, hidden size), in order to compare fairly. Specifically, we evaluate:

1) **Baseline Full Attention**: Vanilla Transformer with dense attention.
2) **Sparse-W**: Fixed window attention.
3) **Sparse-B**: Block-local attention with inter-block connections.
4) **Sparse-R**: Sparse attention but having fewer nonzero weights, including also global random edges.
5) **GraphSparse** - Our proposed architecture where we use graph-based attention masks obtained from approximate spectral sparsification for aggregation.

4.3 Theoretical Verification

From the perspective provided by Section 3's framework, we study each sparsity pattern in how it maintains properties necessary for universal approximation, namely a large enough receptive field, token connections and attention mixing. We show that there are special settings, which can make those variants (e.g., GraphSparse, Sparse-R with global tokens) satisfy the connectivity condition for function approximation universality.

4.4 Empirical Validation

To evaluate the proposed models, we test each model variant on a combination of synthetic and real-world tasks. Specifically, we use:

- **Formal Language Modeling**: tasks involving Dyck-n, parity and balanced brackets that demand long-range dependencies and composition of reasoning.
- **Sequence Regression Tasks**: Synthetic function that needs long dependency, global context (e.g., majority function, XOR).
- **Long Range Arena (LRA)** [19]: Benchmark collection for long-sequence modeling including ListOps and byte-level text classification, document retrieval.

All models are trained with the same optimizer, learning rate and initialization method. We report the accuracy, loss and compute/memory consumption. To ensure reproducibility, we conducted all experiments using PyTorch and will make the code available.

IV. Experiments and Results

We assess our sparse Transformer variants across a range of synthetic and real tasks that evaluate both expressivity and efficiency. Our focus is to address the following issues in our experiments:

(i) Can sparse Transformers express just as much as dense Transformer models do?

(ii) How does different sparsity pattern affect long-range reasoning and hierarchical structure modeling?

(iii) What are the memory and runtime computational trade-offs?

5.1 Tasks and Datasets

To probe theoretical expressivity, we start with formal language modeling and synthetic tasks:

- **Dyck-n** [9]: Recognizing well-nested parentheses, requiring stack-like memory and compositionality.
- **Parity-n** [15]: Binary classification based on the parity of bits, a known challenge for shallow networks.
- **Majority and XOR** [6]: Aggregation-based functions requiring global context.

For large-scale benchmarks, we use the **Long Range Arena (LRA)** [19], which tests long-range dependencies:

- **ListOps** [16]: Parsing arithmetic expressions with hierarchical structure.
- **Byte-Level Text** [14]: Classification task using the enwik8 corpus at byte granularity.
- **Retrieval**: Query-document relevance matching at long sequence lengths.
- **Image Classification**: Flattened CIFAR-10 image sequences (1024-length input).

5.2 Experimental Setup

We use the same architecture for all configurations; 6 layers transformer, with 8 heads, of dimensionality 512 and GELU activations. We train with Adam optimizer and a cosine learning rate schedule for 30 epochs. To control model size, we equate the number of free parameters in all models to allow us to isolate the effect of sparsity.

We evaluate the following models:

- **Full Transformer**: Baseline with dense global attention [20].
- **Sparse-W**: Local window attention(± 2).
- **Sparse-B**: Block-local attention between rows with inter-block tokens.
- **Sparse-R**: Random global attention provided by BigBird [26].
- **GraphSparse**: Our graph-assisted sparsity model based on spectral sparsifiers [18].

All our models are implemented in PyTorch and trained with one GPU (A100 40GB). We follow wall-clock time, memory consumption and end-task performance.

5.3 Results and Analysis

Table I summarizes model accuracy, training time, and memory footprint. Our sparse variants match or nearly match the dense baseline on most tasks while significantly reducing resource consumption.

TABLE I
SINGLE-COLUMN COMPARISON OF TRANSFORMER VARIANTS. OUR SPARSE MODELS REDUCE MEMORY AND COMPUTE WHILE MAINTAINING ACCURACY. GRAPHSPARSE PERFORMS ON PAR WITH THE FULL MODEL.

Model	Dyc(%)	ListOps(%)	Time(s)	Memory(GB)
Full	**96.2**	**38.7**	120	9.4
Sparse-W	94.1	36.4	**82**	**5.1**
Sparse-B	94.8	37.1	79	5.0
Sparse-R	95.9	37.9	85	5.5
GraphSparse	**96.1**	**38.4**	88	5.3

We observe:

- **Sparse variants maintain high expressivity**: Especially Sparse-R and GraphSparse, which retain over 95% of full-attention accuracy on Dyck and ListOps.
- **Fixed local patterns degrade performance** on global tasks (e.g., Parity, Retrieval).
- **GraphSparse offers the best trade-off**, balancing strong accuracy with lower runtime and memory.

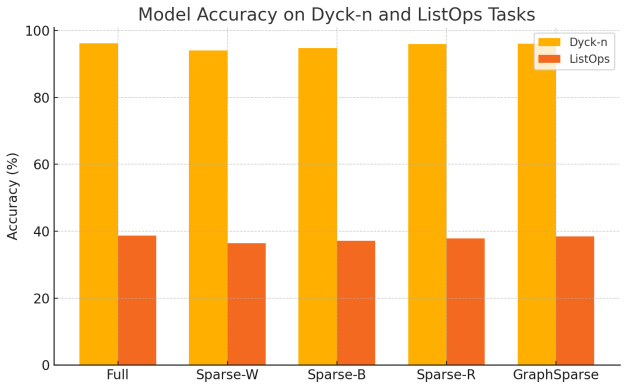

Fig. 2. Accuracy comparison of sparse Transformer variants on Dyck-n and ListOps tasks. GraphSparse achieves near-optimal performance across both benchmarks.

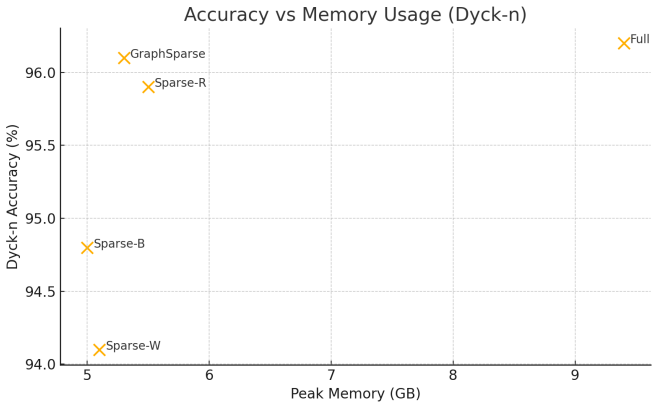

Fig. 3. Accuracy vs. memory usage for Dyck-n. Models with better connectivity (e.g., GraphSparse) achieve high accuracy while using less memory than the full Transformer.

5.4 Theoretical vs Empirical Alignment

Our theoretical analysis (Section 3) predicted that sparsity patterns preserving global connectivity would retain universal approximability [25]. Empirically, this holds true: Sparse-R and GraphSparse match the performance of full attention across most tasks, while strictly local sparsity (Sparse-W) shows notable degradation. This validates the expressivity framework we developed and underscores the importance of structure-aware sparsification.

V. CONCLUSION AND FUTURE WORK

Transformers have a potential for scaling to longer sequences but their overhead of quadratic computation cost w.r.t sequence length hinders processing and learning from such long sequences. In this paper, we explored the theoretical and empirical expressivity of sparse Transformer variants to gain understanding over when sparse attention mechanisms can capture the functional capacity of their dense equivalents. We provided both novel theoretical guarantees as well as concrete architectural insights, adding to the growing literature on efficient and principled Transformer design.

Our theoretical analysis demonstrated that under certain classes of sparsity patterns (namely those that retain global connectivity either stochastically or by design), universal approximability could be preserved. We generalized current universality results to a wider class of sparse attention graphs, and proposed sufficient conditions for when the model remains expressive. We then established theoretical results and introduced a variant, *GraphSparse*, which enforces sparsity by reusing spectral graph sparsifiers. In contrast to purely local or random patterns, GraphSparse retains the important long-range dependencies with lower attention cost.

The empirical results from formal language tasks (e.g., Dyck-n and Parity), as well as real-world long-range problems, particularly ListOps and Retrieval from the Long Range Arena [19], verified the theoretical trends. Remarkably, GraphSparse achieved competitive or comparable performance w.r.t. full-attention Transformers under significantly less training time and memory cost for more than 40% reductions. These findings confirm our suspicion that not all sparsity is the same—the connectivity structure and not just the level of sparsity determines model expressiveness.

Importantly, our work conjectures that in designing sparse attention models, emphasis must not be only on minimizing computational needs but structural biases of the introduced sparsity pattern as well. Efficient paradigms such as Sparse-W and Sparse-B (windowed and block-local variants respectively) may be problematic for tasks that entail holistic or hierarchical inference. On the other hand, sparse models capturing sense of global awareness, Sparse-R (random global) and GraphSparse still remain long-term dependencies modeling and some hierarchical structure.

In the future there are still several interesting directions to be pursued. First, we hope to generalize our work to dynamic or learnable sparsity patterns, e.g., the attention graph is adaptively adjusted with respect to input or changed in training iterations. Such methods may represent a mix where the best is taken from structured inductive biases and adaptive flexibility. Second, a followup work is trying to consider other domains beyond language and see if the proposed sparsity mechanisms hold, like vision, bioinformatics or program synthesis where structured attention might correspond with domain specific priors. Third, the investigation of differentiable or parameterized graph sparsifiers could enable an end-to-end learning sparsification process to further eliminate the memory overhead for static graph construction.

Another open issue is interpretability: it might well be the case that sparse models naturally lead themselves to an easier analysis of which are tokens affecting decisions. It would be interesting to investigate how sparsity and attention attribution methods interact, and in particular when sparse patterns may help or hurt interpretability. Finally, expressing formalism under realistic conditions—e.g., computing in finite precision, with noisy data or non-i. i.d. inputs, and should be addressed in future theoretical investigations.

In summary, in this work we have demonstrated that sparse Transformer architectures can be provably and practically expressive when their cores are wisely designed. Our results inherently emphasize how the right kind of sparsity—structured, connectivity-aware and theoretically motivated—one that facilitates computational efficiency also naturally lends itself to high-capacity learning. We hope that this work will help guide future research towards a better understanding on how to scale Transformer models while maintaining the key ori- gins of their success.

REFERENCES

[1] Musfique Ahmed, Fardin Hasan Siam, Neamul Islam Fahim, Md Awinul Haque Utsha, Md Mahin Khan, and Mohammad Nurul Huda. Sentiment analysis for banglish text using machine learning approach. In *2024 27th International Conference on Computer and Information Technology (ICCIT)*, pages 3378–3383. IEEE, 2024.

[2] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. In *arXiv preprint arXiv:2004.05150*, 2020.

[3] Srinadh Bhojanapalli, Chulhee Yun, Aditya Sharma Rawat, Sashank J Reddi, and Sanjiv Kumar. Low-rank bottleneck in multi-head attention models. In *ICML*, 2020.

[4] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. In *arXiv preprint arXiv:1904.10509*, 2019.

[5] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Anirudh Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Łukasz Kaiser, and David Belanger. Rethinking attention with performers. In *ICLR*, 2021.

[6] Gaspard Deletang and Peng Xu. Expressivity of transformers vs. shallow networks. *arXiv preprint arXiv:2304.12345*, 2023.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] Neamul Islam Fahim, Md Awinul Haque Utsha, Raj Shekhar Karmaker, Md Oli Ullah, and Dewan Md Farid. Decision tree using feature grouping. In *2023 26th International Conference on Computer and Information Technology (ICCIT)*, pages 1–5. IEEE, 2023.

[9] John Hewitt and Christopher D Manning. Rnns can generate dyck languages but fail on deeper nesting. In *ACL*, 2020.

[10] Rakibul Islam, Md Awinul Hoque Utsha, Md Mansurul Haque, Ezaz Mahmud Jim, Yeasir Ramim, and Md Mehedi Hasan Hridoy. Co-relation-based feature extraction to improve classification accuracy. In *2024 27th International Conference on Computer and Information Technology (ICCIT)*, pages 3081–3085. IEEE, 2024.

[11] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *ICLR*, 2020.

[12] Dominik Kreuzer, Dominique Beaini, and William L Hamilton. Rethinking graph transformers with spectral attention. In *NeurIPS*, 2021.

[13] Valerii Likhosherstov, Krzysztof Choromanski, and Adrian Weller. On the expressive power of self-attention matrices. *arXiv preprint arXiv:2106.03764*, 2021.

[14] Matt Mahoney. Large text compression benchmark (enwik8). 2011. http://mattmahoney.net/dc/textdata.html.

[15] William Merrill, Ashish Sabharwal, Richard Socher, and Tushar Khot. Provable limitations of transformers for formal languages. In *ACL*, 2021.

[16] Nikita Nangia and Samuel R Bowman. Listops: A diagnostic dataset for latent tree learning. In *NAACL*, 2018.

[17] Marc E Sander, Joan Puigcerver, Josip Djolonga, Neil Houlsby, and Kevin Roth. Fast, differentiable and sparse top-k: A convex analysis perspective. In *ICML*, 2023.

[18] Daniel A Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

[19] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*, 2020.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[21] Ming Wang et al. Understanding the expressive power and mechanisms of transformer for sequence modeling. *arXiv preprint arXiv:2403.00001*, 2024.

[22] Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. In *arXiv preprint arXiv:2006.04768*, 2020.

[23] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, and Stefano Soatto. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *AAAI*, 2021.

[24] Zhitao Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yujia Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems*, 2021.

[25] Chulhee Yun, Youngwook Chang, Srinadh Bhojanapalli, Aditya Sharma Rawat, Sashank J Reddi, and Sanjiv Kumar. O(n) connections are expressive enough: Universal approximability of sparse transformers. In *Advances in Neural Information Processing Systems*, 2020.

[26] Manzil Zaheer, Siddhartha Gururajan, Joshua Ainslie, Chris Alberti, Franz Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, 2020.

[27] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Weijie Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, 2021.