Recent Progress in Understanding Transformer and Developing Its Surpasser: A Survey

M. Liu¹, D. Yang,² F. R. Beyette Jr.¹, S. Wang³, L. Zhao³

1. School of Electrical and Computer Engineering, University of Georgia, Athens, Georgia, USA

2. Institute for Artificial Intelligence, University of Georgia, Athens, Georgia, USA

3. School of Math. and Info. Tech., Hebei Normal University of Sci. & Tech., Qinhuangdao, Hebei, China {maohua.liu, dy77706, fred.beyette}@uga.edu, wangshi2665@hevttc.edu.cn, zhao liqiang@126.com

Abstract—Transformers have significantly impacted machine learning, particularly in natural language processing and computer vision, due to their robust attention mechanisms and scalability. Although it is very successful and has the attention mechanism to explain it, it is still a black box in general. This lack of transparency is not conducive to applications, nor is it conducive to the development of more advanced models. We systematically analyze current methodologies for interpreting the attention patterns, hidden representations, and decisionmaking processes within Transformers. Additionally. we investigate how these insights aid in refining Transformer architectures and inspire the creation of innovative models that extend beyond the conventional Transformer frameworks. Due to its expertise in long-term dependencies and data scalability, Transformer undoubtedly has great advantages in foundation language models, achieving multimodality, and processing discrete data, but it is inferior to Mamba in terms of efficiency, especially when processing continuous data. It is inferior to Diffusion Model when processing image data because it is not good at grasping the distribution of high-dimensional data and is relatively less suitable for image generation model. By integrating achievements in both explaining and advancing Transformer-like architectures, this paper serves as a valuable resource for researchers aiming to enhance the performance, transparency, and efficiency of Transformers in various applications and to develop models that surpass current Transformer paradigms.

Keywords— Transformer, explainability, Interpretability, Mamba, Diffusion Model.

I. INTRODUCTION

Introduced by Vaswani et al. in 2017 [1], Transformers have become a cornerstone in the field of machine learning, revolutionizing various applications including natural language processing (NLP) [1], computer vision (CV) [2], time series (TS) [3], speech processing [4], bioinformatics [5], graph data [6], music generation [7], robotics [8], symbolic mathematics [9], recommend systems [10], 3D point cloud [11], financial forecasting [12] and so on. As shown in Figure 1, this is an approximation of different applications of Transformer based on our search. The key innovation behind Transformers lies in their attention mechanism, which allows them to find the long-term dependencies of input data in parallel mode. It is well suited for finding longterm dependencies among large and complex sets of discrete data points.



Transformer's main applications



Figure 1. Transformer's main applications.

Despite their widespread adoption and success, the inner workings of Transformers remain largely opaque, often described as "black-box" models. This opacity presents significant challenges for researchers and practitioners who seek to understand, trust, and improve these models. As Transformers continue to be deployed in critical applications, from automated customer service to medical diagnosis, the need for transparency and interpretability becomes increasingly urgent [13][14][15][16].

Recent research has focused on elucidating the mechanisms behind Transformers, aiming to make their operations more interpretable [17]. Various methods have been developed to visualize and understand attention patterns, analyze hidden representations, and decode the decision-making processes within these models [18][19][16]. These efforts are not only vital for validating model outputs but also for guiding the design of more efficient and effective Transformer architectures [2][20].

This paper aims to provide a comprehensive overview of the recent progress in explaining Transformer models. We systematically examine the state-of-the-art methodologies for interpreting these models and highlight how these insights contribute to the refinement of existing architectures and the creation of new ones. By synthesizing the achievements in both explaining and evolving Transformer architectures, this paper seeks to offer a valuable resource for researchers dedicated to enhancing the performance, transparency, and efficiency of these models in a wide array of applications.

In the subsequent sections, we delve into the various approaches for interpreting Transformer models, discuss the advancements beyond traditional Transformer architectures, and explore the implications of these developments for the future of machine learning. Through this comprehensive survey, we hope to shed light on the intricate workings of Transformers and inspire further innovations in this dynamic and rapidly evolving field. Specifically, Section II describes the architecture and mechanism of Transformer and its comparison with RNN and CNN; Section III covers recent advances in understanding Transformer and developing its transcendent; Section IV summarizes the paper.

II. TRANSFORMER ARCHITECTURE AND MECHANISM

The Transformer architecture has fundamentally changed the landscape of deep learning by employing selfattention mechanisms, which significantly enhance the ability to model long-term dependencies in sequential data. This section details the key components and mechanisms that constitute the Transformer architecture and compare it to the other architectures, providing insights into its operation and effectiveness.

A. Transformer Architecture

The Transformer architecture is built around an encoderdecoder structure, with both the encoder and decoder consisting of multiple identical layers. Each layer has two primary components: a multi-head self-attention module and a position-wise feed-forward network. The encoder processes the input sequence into continuous representations, while the decoder generates the output sequence by using the encoder's representation and previously generated tokens. The decoder layers also have an additional sub-layer to handle encoder-decoder attention [1].

The Multi-Head Attention module allows the model to focus on different parts of the input sequence simultaneously by employing multiple attention heads. Each head performs scaled dot-product attention, which computes attention scores by taking the dot product of query (Q) and key (K) vectors, scaling by the square root of the dimension of the key vectors, and applying a *softmax* function.

The Feed-Forward Network is a fully connected network applied to each position separately and identically. It consists of two linear transformations with a ReLU



Figure 2. The architecture of the Transformer model [1].

activation in between.

The attention mechanism allows each position in the input sequence to attend to all positions, enabling the model to capture long-term dependencies without regard to the distance between positions [1][15]. The self-attention mechanism operates as follows: Compute the dot products of the query with all keys; Scale the dot products by the square root of the key dimension; Apply the *softmax* function to obtain attention weights; Multiply the attention weights by the value vectors to obtain the weighted sum. Mathematically, the attention mechanism can be defined as below:

Attention
$$(Q, K, V) = softmax \left(\frac{QK^T}{\sqrt{d_k}}\right) V$$
 (1)

where Q (queries), K (keys), and V (values) are matrices packed together, and d_k is the dimension of the keys [1].

Since Transformers do not have a built-in mechanism for capturing the order of tokens, positional encodings are added to the input embeddings. These encodings allow the model to differentiate between the positions of tokens in a sequence and are crucial for processing sequential data.

B. Comparison with other architectures

The Transformer model's architecture and mechanisms

Architecture	Transformer [1][2][4]	RNN [21][22][30][31]	CNN [23][26][27][32]
Basic structure	Self-attention layers, feed- forward neural networks, positional encoding	Sequential processing with recurrent connections for temporal data	Convolutional layers for spatial data processing
Parallelization	Highly parallelizable, processes entire sequences parallelly	Sequential processing, making it less parallelizable	Highly parallelizable, processes multiple parts of the data simultaneously
Long-term dependencies	Efficiently captures long-term dependencies through self-attention	Struggles with long-term dependencies due to vanishing gradient problem	Effective in capturing local features but less suited for long-term dependencies
Training efficiency	Efficient, especially on large datasets, thanks to parallel processing	Slower training due to sequential nature	Faster training due to parallel processing and efficient use of local patterns
Data type	Primarily sequential data (text, time series, speech), but adaptable for images and multimodal data	Sequential data (text, speech, time series)	Primarily spatial data (images, videos), but adaptable to text and sequential data
Advantages	Can handle long-term dependencies efficiently, Highly parallelizable, Scales well with large datasets	Maintains sequence order, Good for temporal data, Suitable for tasks like language modeling	Excellent for spatial feature, Parallelizable and efficient, Effective in recognizing patterns in images
Disadvantages	Resources consuming, Can be data-intensive, May need large datasets for effective training	Struggles with long-term dependencies, Slower training, Vanishing/exploding gradients	Limited in handling long- term dependencies, Not as efficient for sequential data

Table	1:	Com	parison	of	Transformer.	RNN	and	CNN
1 uore	т.	Com	parison	O1	runsionner,	1/1 /1 /	unu	

have set new benchmarks in multiple machine learning tasks due to their ability to model long-term dependencies and parallelize training. By comparing it to the other architectures like RNN and CNN, we can know more of its characters, advantages and disadvantages, as shown in Table 1.

Transformer is very different from RNN and CNN in structure and principle, so it has great differences and advantages. It perfectly replaces RNN in NLP and is the optimal solution. But when processing other continuous signals, especially when it is necessary to master the distribution of data, RNN still has certain advantages. At the same time, it performs much better than CNN in mastering long-term dependencies, but is still not as good as CNN in mastering local data distribution. CNN still shows advantages in mastering spatial features. This is why there are many studies trying to combine the advantages of them. Although Transformer is gaining more and more attentions, CNN/RNN have not been forgotten. CNN/RNN are simpler and computationally more efficient compared to Transformer. They do not require complex mechanisms like attention and are easier to implement and train on standard hardware [24]. They can be more efficient in handling specific types of data and tasks [25]. Recently, CNN/RNN have been used for building generative/foundation models. They are not only used to assist Transformer [26] [27] but also work alone in foundation models [28] [29] [30] [31] [32].

III. UNDERSTAND AND SURPASS TRANSFORMER

Despite demonstrating significant practical efficacy and being partially explicable through the attention mechanism, the Transformer remains somewhat opaque. This opacity arises from various factors: the intricate complexity of its layers and parameters, the nonlinear and intertwined nature of its operations, the highdimensional representations it employs, the dynamic nature of its training process, and its reliance on learned rather than explicit rules. Consequently, achieving precise control, adjustment, transplanting and enhancement of the Transformer poses considerable challenges.

There have been many efforts to explain transformers. According to the characteristics of these efforts, we divide them into three categories: primary stage, intermediate stage and advanced stage. Also, we will review some existed architectures similar to or beyond Transformer.

A. Primary stage.

The biggest feature of this stage is to use existing explanation AI (XAI) techniques to explain Transformer. There have been some literature reviews on this [33] [34] [35] [36]. These methods mainly include following:

Activation-Based Methods. Activation-based methods help in understanding the internal workings of Transformers by analyzing neuron activations. These methods identify which neurons are activated by specific input features, providing insights into how the model processes information [37][38].

Gradient-Based Methods. Gradient-based techniques, such as Integrated Gradients and Layer-wise Relevance Propagation (LRP), attribute the model's predictions to its input features by computing gradients. Voita et al. demonstrated that specific attention heads in Transformers are crucial for capturing syntactic and long-term dependencies, which is revealed through gradient-based attribution [39]. Additionally, Jain and Wallace discussed the limitations of attention as a sole explanation, highlighting the need for gradient-based methods to provide more reliable interpretations [40].

Pruning Techniques. Pruning techniques involve removing parts of the model, such as specific attention heads, to analyze their impact on performance. This helps identify the importance of different components within the Transformer. Voita et al. used pruning to show that certain attention heads can be removed without significantly affecting the model's performance, suggesting that only a subset of the heads are essential for specific tasks [39]. This approach helps in simplifying the model and improving interpretability.

Perturbation-Based Methods. Perturbation-based methods explain model predictions by altering input features and observing the changes in output. Techniques like LIME (Local Interpretable Model-agnostic Explanations) perturb input features to determine their importance in the model's decision-making process [4]. Serrano and Smith used perturbation to show that removing high-attention words does not always impact model predictions significantly, raising questions about the reliability of attention-based explanations alone [42].

One disadvantage of the primary stage is post-doc. posthoc interpretability refers to methods applied after the model has been trained to explain its predictions. This approach is often criticized for potentially providing misleading explanations since the explanations are not inherently tied to the model's decision-making process.

Page 4 of 8

B. Intermediate stage.

The biggest feature of this stage is to use the attention mechanism to explain Transformer. Attention-based methods have become central to understanding and explaining the behavior of Transformer models. Attention-based methods mainly include following:

Visualization of Attention Weights. One of the primary methods to interpret Transformers is by visualizing attention weights. These visualizations help understand how the model focuses on different parts of the input during processing. For instance, tools like BertViz [42] provide interactive visualizations of attention heads in BERT models, revealing patterns in how attention is distributed across different layers and heads.

Attribution Methods. Attention attribution methods, such as Layer-wise Relevance Propagation (LRP) and Integrated Gradients, aim to attribute the model's predictions to its input features. Voita et al. [39] analyzed attention heads in Transformers and found that specific heads are responsible for syntactic functions, while others capture long-term dependencies. Such attributions offer insights into the internal workings of the model and help identify the roles of different attention heads.

Self-Explanation Models. Another approach involves self-explaining models where the model itself generates explanations for its predictions. This is achieved by adding interpretability constraints during training. Jain and Wallace [40] argued that while attention provides some interpretability, it does not always correlate with feature importance, suggesting the need for more robust self-explaining mechanisms.

Use of Probing Tasks. Probing tasks are designed to investigate what linguistic information is captured by different layers and attention heads in Transformers. Hewitt and Manning [43] used structural probes to show that Transformers encode syntactic tree structures in their representations, providing a deeper understanding of the model's linguistic capabilities.

Despite their usefulness, attention-based explanations have faced criticism for their lack of consistency and reliability. Serrano and Smith [16] demonstrated that removing high-attention words does not always significantly impact the model's predictions, questioning the direct interpretability of attention weights. Consequently, alternative methods like SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) have been proposed to complement attention-based explanations [41].

C. Advanced stage.

This stage emphasizes on white-box Transformer and, mathematical interpretability.

Recent research has introduced the Coding Rate Transformer (CRATE), a white-box transformer architecture designed for efficient data compression and sparsification. CRATE models utilize a Sparse Rate Reduction (SRR) objective, which transforms input data into structured and compact representations. This method involves an alternating optimization procedure that incrementally improves data representations by compressing and sparsifying them at each layer [44][45]. CRATE consider Transformer as a model to compress and transform distributions iteratively. Specifically, the attention module is used to transform the distribution while the forward-feedback module is used to compress the distribution [45]. There are several key features in CRATE.

Sparse Rate Reduction. CRATE employs SRR to measure the quality of learned representations. This involves minimizing the number of bits required to encode data while promoting sparsity, leading to more compact and interpretable features [45].

Iterative Optimization: Each layer of the CRATE model performs a step of the optimization algorithm, akin to proximal gradient descent, to improve data compression and sparsification. This method ensures that each layer contributes to refining the data representation [45].

Mathematical Interpretability. Unlike traditional black-box models, CRATE provides a clear mathematical framework, making each layer's function and the overall model architecture interpretable. This transparency helps in understanding how the model processes and transforms data [45].

Competitive Performance. Despite being simpler than conventional transformers like ViT, BERT, and GPT-2, CRATE has shown competitive performance across various tasks, including image classification, unsupervised masked completion, and self-supervised feature learning. It achieves similar accuracy with fewer parameters, demonstrating its efficiency [45].

Scalability and Flexibility. CRATE has proven effective on large-scale real-world datasets, both supervised and unsupervised, and in various applications such as image and text data. This versatility highlights its potential as a robust foundation model in diverse AI tasks [45].

CRATE represents a significant advancement in the development of white-box transformers, combining

efficiency, interpretability, and competitive performance. Its focus on data compression and sparsification, coupled with a robust mathematical foundation, positions it as a promising alternative to traditional transformer models in various AI applications.

D. Similar advanced architectures

As the understanding of Transformer increases, researchers are also trying to develop other advanced models similar to Transformer, such as Mamba, Diffusion model and Diffusion Transformer.

Mamba. Mamba is a State Space Model (SSM) developed to enhance the efficiency of Transformer. Mamba optimized Transformer's self-attention mechanism to a near-linear computational complexity. Mamba can model long-sequence dependencies like Transformer but with near-linear computational cost, leading to significantly higher speed and efficiency.

Mamba does not utilize the same attention mechanism with Transformer. Its core concept is based on compressing continuous signals using orthogonal polynomials [46], with the "state space" referring to the space constructed from dimensions corresponding to the orthogonal polynomial bases. This gives Mamba an advantage in modeling continuous data, as all orthogonal polynomial bases are continuous [47] [48] [49].

Since 2024, there has been increasing interest in using Mamba for time series analysis. Researchers have published approximately four papers on this topic, including "Is Mamba Effective for Time Series Forecasting?" [50], "TimeMachine: A Time Series is Worth 4 Mambas for Long-term Forecasting" [51], "MambaStock: Selective State Space Model for Stock Prediction" [52], and "SiMBA: Simplified Mamba-Based Architecture for Vision and Multivariate Time Series" [53]. These studies consistently demonstrate that Mamba outperforms previous models in both accuracy and efficiency for time series analysis.

Diffusion model (DM). As a generative model, DM creates high-quality data samples by reversing a diffusion process. They start with noise and iteratively denoise the data through a sequence of steps. Each step refines the data by conditioning on the previous one, making the process gradual and controlled [54].

From the perspective of generative model, DM shows great advantage over Transformer in transforming and compressing distribution, capturing fine details [55], training stability [56], handling complex data distribution [56], computational cost [1] and data reliance [57]. DM offers significant advantages over Transformers in generating images, primarily due to their iterative refinement process, ability to capture fine details, training stability, and effectiveness in handling complex

Architecture	Transformer [42][43][44][45]	Mamba [46][47][48][53]	Diffusion Model [54][55][57]	Diffusion Transformer [58]	
Basic Structure Self-attention layers, feed-forward neural networks, positional hencoding C		Combines self-attention mechanisms with hierarchical design, focusing on efficiency	Sequential process of adding and removing noise to generate data	Integrates diffusion processes with self- attention mechanisms for sequential data	
Mechanism	Iechanism Self-attention to capture dependencies across entire sequences entire sequences for the second se		Stepwise diffusion, starting from noise to create or denoise samples	Combines diffusion (probabilistic modeling) with attention for efficient data generation	
Parallelization	Highly parallelizable, processes entire sequences at once	Designed for better parallel processing and scalability	Not inherently parallel; sequential processing for diffusion steps	Combines parallelizable elements of Transformers with sequential steps of diffusion	
Applications	NLP, Computer Vision, Time Series, Multimodal tasks	Optimized for fast processing of large-scale data	Image and data generation, noise removal, generative tasks	Advanced generative tasks, combining benefits of Transformers and Diffusion Models	
Long-term Dependencies	g-term endencies Efficiently captures long-term dependencies Improved handling of via self-attention		Effective for structured data but not designed for dependencies	Captures dependencies while maintaining generative quality through diffusion	
Advantages	Can handle long-term dependencies efficiently, Highly parallelizable, Scales well with large datasets,	Optimized for speed and resource efficiency, Scalable for large tasks, Reduced computational cost	Effective for generating high- quality data, Excellent at capturing fine-grained details	Combines strengths of both Transformer and Diffusion models, Maintains generative power with efficient sequence processing	
Disadvantages	Requires significant computational resources, Can be data-intensive, May need large datasets for effective training	Newer model with less extensive testing across diverse tasks, May need fine-tuning for specific use cases	Sequential nature slows down training and sampling, Resource-intensive for high-quality output	Combines complexity of both Transformers and Diffusion, making it more resource-intensive	
Excels In	NLP (e.g., GPT, BERT), Computer Vision (e.g., Vision Transformer), Time Series (e.g., Informer)	Optimized for large-scale data processing, Applications requiring fast attention computation, Potential use in NLP and large data tasks	Image Generation (e.g., DALL·E 2) Speech Synthesis, Data Denoising, Generative Art	Complex generative tasks, Data generation across multiple domains (e.g., multimodal tasks), Advanced language modeling combined with generation	

Table 2:	Comparison	between	Transformer	and its	s surpassers
----------	------------	---------	-------------	---------	--------------

data distributions. These strengths make DM a superior choice for high-quality and detailed image generation.

As shown in Table 2, Mamba not only has the advantages of Transformer but is more efficient and has a more obvious advantage in processing continuous data. DM is an excellent generative model, especially when facing high-dimensional data such as images, it surpasses other models in both efficiency and accuracy. This is because DM has an unparalleled advantage in grasping the distribution of images. This is why people try to combine Transformer with DM to give birth to Diffusion Transformer.

IV. SUMMARY

Transformers have revolutionized machine learning, particularly in natural language processing and computer vision, due to their powerful attention mechanisms and scalability. However, the complexity of Transformer architectures poses challenges in understanding their internal workings. This review paper provides a comprehensive survey of recent advancements aimed at demystifying the "black-box" nature of Transformers. We systematically analyze methodologies for interpreting attention patterns, hidden representations, and decision-making processes within Transformers, including visualization techniques, attribution methods, and probing tasks.

Additionally, we explore recent progress in developing Transformer-like models based on similar or very different mechanisms and architectures, which offer competitive performance and simpler interpretability compared to Transformers. These models leverage the strengths of their respective architectures, such as the efficiency of convolutions and the temporal modeling capabilities of RNNs, to achieve state-of-the-art results in various tasks.

By integrating achievements in explaining and advancing Transformer architectures, this paper serves as a valuable resource for researchers. It aims to enhance the performance, transparency, and efficiency of Transformers in various applications and to inspire the development of innovative models that extend beyond the conventional Transformer frameworks.

REFERENCES

- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," *Advances in Neural Information Processing Systems* 30, 2017.
- [2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," arXiv preprint arXiv:2010.11929, 2020.
- [3] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond Efficient Transformer for Long Sequence Time-series Forecasting," *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 12, pp. 11106-11115, May 2021.
- [4] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han et al., "Conformer: Convolution-augmented Transformer for Speech Recognition," arXiv preprint arXiv:2005.08100, 2020.
- [5] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, and A. Bridgland, "Highly Accurate Protein Structure Prediction with AlphaFold," *Nature*, vol. 596, no. 7873, pp. 583-589, 2021.
- [6] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T. Y. Liu, "Do Transformers Really Perform Badly for Graph Representation?," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 28877-28888, 2021.
- [7] C. Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music Transformer," *arXiv preprint arXiv:1809.04281*, 2018.
- [8] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision Transformer: Reinforcement Learning via Sequence Modeling," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 15084-15097, 2021.
- [9] G. Lample and F. Charton, "Deep Learning for Symbolic Mathematics," *arXiv preprint arXiv:1912.01412*, 2019.
- [10] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "BERT4Rec: Sequential Recommendation with Bidirectional

Encoder Representations from Transformer," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1441-1450.

- [11] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point Transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 16259-16268.
- [12] C. Zhang, N. N. A. Sjarif, and R. Ibrahim, "Deep Learning Models for Price Forecasting of Financial Time Series: A Review of Recent Advancements: 2020–2022," *Wiley Interdiscip. Rev.: Data Min. Knowl. Discov.*, vol. 14, no. 1, p. e1519, 2024.
- [13] W. Samek and K. Müller, "Towards Explainable Artificial Intelligence," *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, W. Samek, G. Montavon, A. Vedaldi, L. Hansen, and K. Müller, Eds. Cham: Springer, 2019, pp. 5–22.
- [14] Z. Lipton, "The Mythos of Model Interpretability," *Communications of the ACM* 61, 2016, pp. 36–43.
- [15] J. Devlin, M. Chang, K. Lee and K. Toutanova, "BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding," North American Chapter of the Association for Computational Linguistics, 2019.
- [16] S. Serrano, and N. Smith, "Is Attention Interpretable?," arXiv preprint arXiv:1906.03731, 2019.
- [17] Y. Belinkov and J. Glass, "Analysis Methods in Neural Language Processing: A Survey," *Transactions of the Association for Computational Linguistics* 7, 2018, pp. 49–72.
- [18] K. Clark, U. Khandelwal, O. Levy, and C. Manning, "What does Bert Look at? An Analysis of Bert's Attention," *arXiv preprint* arXiv:1906.04341, 2019.
- [19] J. Vig and Y. Belinkov, "Analyzing the Structure of Attention in a Transformer Language Model," arXiv preprint arXiv:1906.04284, 2019.
- [20] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *OpenAI Blog*, 2018 (available at: <u>https://openai.com/index/language-unsupervised/</u>).
- [21] K. Cho, "Learning Phrase Representations Using RNN Encoderdecoder for Statistical Machine Translation," arXiv preprint arXiv:1406.1078, 2014.
- [22] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541-551, 1989.
- [23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," Adv. Neural Inf. Process. Syst., vol. 25, 2012.
- [24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, Boston: MIT Press, 2009.
- [25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature* 521, No. 7553, 2015, 436–444.
- [26] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11976–11986.
- [27] B. Lim, S. Arık, N. Loeff, and T. Pfister, "Temporal Fusion Transformers for Interpretable Multi-Horizon Time Series Forecasting," *International Journal of Forecasting* 37, No. 4 2021, pp. 1748–1764.
- [28] V. Ekambaram, A. Jati, N. Nguyen, P. Sinthong, and J. Kalagnanam, "TSMixer: Lightweight MLP-Mixer Model for Multivariate Time Series Forecasting," *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 459–469.
- [29] V. Ekambaram, A. Jati, N. Nguyen, P. Dayama, C. Reddy, W. Gifford, and J. Kalagnanam, "TTMs: Fast Multi-Level Tiny Time Mixers for Improved Zero-Shot and Few-Shot Forecasting of Multivariate Time Series," arXiv preprint arXiv:2401.03955, 2024.

- [30] H. Hou and F. Yu, "RWKW-TS: Beyond Traditional Recurrent Neural Network for Time Series Tasks," arXiv preprint arXiv:2401.09093, 2024.
- [31] Z. Yue, Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, and B. Xu, "Ts2vec: Towards Universal Representation of Time Series," In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36, No. 8, 2022, pp. 8980–8987.
- [32] X. Zhang, Z. Zhao, T. Tsiligkaridis, and M. Zitnik, "Self-Supervised Contrastive Pre-Training for Time Series via Time-Frequency Consistency," *Advances in Neural Information Processing Systems* 35, 2022, pp. 3988–4003.
- [33] J. E. Zini and M. Awad, "On the Explainability of Natural Language Processing Deep Models," ACM Comput. Surv., vol. 55, pp. 1-31, 2022.
- [34] R. Kashefi, L. Barekatain, M. Sabokrou, and F. Aghaeipoor, "Explainability of Vision Transformers: A Comprehensive Review and New Perspectives," *arXiv preprint arXiv:2311.06786*, 2023.
- [35] S. Vijayakumar, "Interpretability in Activation Space Analysis of Transformers: A Focused Survey," arXiv preprint arXiv:2302.09304, 2023.
- [36] A. Braşoveanu and R. Andonie, "Visualizing Transformers for NLP: A Brief Survey," *Proceedings of the 2020 24th International Conference Information Visualisation*, 2020, pp. 270–279.
- [37] H. Chefer, S. Gur, and L. Wolf, "Transformer Interpretability beyond Attention Visualization," *Proceedings of the IEEE/CVF* conference on computer vision and pattern recognition, 2021, pp. 782–791.
- [38] M. Muhammad and Mohammed Yeasin, "Eigen-Cam: Class Activation Map Using Principal Components." Proceedings of the 2020 international joint conference on neural networks (IJCNN), 2020, pp. 1–7.
- [39] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov, "Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned," *arXiv preprint* arXiv:1905.09418, 2019.
- [40] S. Jain and B. Wallace, "Attention is Not Explanation," arXiv preprint arXiv:1902.10186, 2019.
- [41] S. Lundberg and S. Lee, "A Unified Approach to Interpreting Model Predictions," Advances in Neural Information Processing Systems 30, 2017.
- [42] J. Vig, "A Multiscale Visualization of Attention in the Transformer Model," arXiv preprint arXiv:1906.05714, 2019.
- [43] J. Hewitt and C. Manning, "A Structural Probe for Finding Syntax in Word Representations," 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), 2019, pp. 4129–4138.
- [44] J. Yang, X. Li, D. Pai, Y. Zhou, Y. Ma, Y. Yu, and C. Xie, "Scaling White-Box Transformers for Vision," arXiv preprint arXiv:2405.20299, 2024.
- [45] Y. Yu, S. Buchanan, D. Pai, T. Chu, Z. Wu, S. Tong, H. Bai, Y. Zhai, B. Haeffele, and Y. Ma, "White-Box Transformers via Sparse Rate Reduction: Compression Is All There Is?," arXiv preprint arXiv:2311.13110, 2023.
- [46] A. Gu, T. Dao, S. Ermon, A. Rudra, and C. Ré, "Hippo: Recurrent Memory with Optimal Polynomial Projections," *Advances in Neural Information Processing Systems*, 33, 2020, pp. 1474– 1487.
- [47] A. Gu and T. Dao, "Mamba: Linear-Time Sequence Modeling with Selective State Spaces," arXiv preprint arXiv:2312.00752, 2023.
- [48] A. Gu, K. Goel, and C. Ré, "Efficiently Modeling Long Sequences with Structured State Spaces," arXiv preprint arXiv:2111.00396, 2021.

- [49] D. Fu,T. Dao, K. Saab, A. Thomas, A Rudra, and C. Ré, "Hungry Hungry Hippos: Towards Language Modeling with State Space Models," arXiv preprint arXiv:2212.14052, 2022.
- [50] Z. Wang, F. Kong, S. Feng, M. Wang, H. Zhao, D. Wang, and Y. Zhang, "Is Mamba Effective for Time Series Forecasting?," arXiv preprint arXiv:2403.11144, 2024.
- [51] M. Ahamed and Q. Cheng, "TimeMachine: A Time Series is Worth 4 Mambas for Long-term Forecasting," arXiv preprint arXiv:2403.09898, 2024.
- [52] Z. Shi, "MambaStock: Selective State Space Model for Stock Prediction," arXiv preprint arXiv:2402.18959, 2024.
- [53] B. Patro and V. Agneeswaran, "SiMBA: Simplified Mamba-Based Architecture for Vision and Multivariate Time Series," *arXiv preprint arXiv:2403.15360*, 2024.
- [54] F. Croitoru, V. Hondru, R. Ionescu, and M. Shah, "Diffusion Models in Vision: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, No. 9, 2023, 10850– 10869.
- [55] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," Advances in Neural Information Processing Systems 33, 2020, 6840–6851.
- [56] Y. Song, J. Sohl-Dickstein, D. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-Based Generative Modeling Through Stochastic Differential Equations," *arXiv preprint arXiv:2011.13456*, 2020.
- [57] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan et al., "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems* 33, 2020, pp. 1877–1901.
- [58] W. Zhao, Y. Han, J. Tang, K. Wang, Y. Song, G. Huang, F. Wang, and Y. You, "Dynamic Diffusion Transformer," *arXiv preprint* arXiv:2410.03456, 2024.