Attention Models in Machine Learning

A Comprehensive Technical Survey of Mechanisms, Architectures, and Applications

Asrat Amnie, Associate Professor, The City University of New York

(unpublished Manuscript)

Abstract

Attention mechanisms constitute one of the most transformative advances in modern machine learning, enabling neural networks to dynamically weight and integrate contextual information rather than relying on fixed-length compressed representations. Originating with additive alignment scoring in recurrent neural machine translation (Bahdanau et al., 2015), the field underwent a paradigm shift with the introduction of the Transformer architecture and scaled dot-product attention (Vaswani et al., 2017). This survey provides a systematic and chronologically ordered taxonomy of attention mechanisms, from their foundational formulations through self-attention, multi-head attention, efficient sparse variants, and domain-specific extensions in vision, graph learning, and multimodal systems. We present the mathematical underpinnings of each mechanism, analyze computational complexity trade-offs, review landmark architectures including BERT (Devlin et al., 2019), GPT-3 (Brown et al., 2020), Vision Transformer (Dosovitskiy et al., 2021), and Graph Attention Networks (Velickovic et al., 2018), and discuss the role of backpropagation and optimization in training attention-based systems. The survey synthesizes over two decades of research to provide a coherent reference for researchers and practitioners working across natural language processing, computer vision, speech recognition, and multimodal AI.

Keywords: attention mechanisms, self-attention, Transformer, multi-head attention, scaled dot-product attention, efficient attention, vision transformer, graph attention, cross-attention, machine learning

1. Introduction

In contemporary machine learning, particularly within deep learning and sequence modeling, attention mechanisms constitute a family of architectures that dynamically weight information according to contextual relevance. Rather than compressing all input information into a single fixed-

length vector, attention architectures compute relevance between each element of a query and all elements of a key-value store, enabling selective focus on the most pertinent portions of the input.

The significance of attention extends far beyond neural machine translation, where it first achieved prominence. Today, attention underlies state-of-the-art systems in natural language processing (NLP), computer vision, speech recognition, reinforcement learning, and multimodal AI. The defining architectural transition occurred with the Transformer (Vaswani et al., 2017), which dispensed entirely with recurrence and convolution, positioning attention as the central computational primitive. Subsequent years witnessed an explosion of attention-based architectures addressing challenges ranging from quadratic scaling to domain-specific inductive biases.

This survey is organized chronologically and conceptually. We begin with the historical origins of attention in sequence-to-sequence learning (Section 2), proceed through foundational mechanisms and their mathematics (Sections 3–5), cover the Transformer architecture and its derivatives (Sections 6–8), survey efficient and sparse variants (Section 9), and conclude with domain-specific extensions and the training dynamics of attention-based models (Sections 10–14). A comprehensive reference list of peer-reviewed sources is provided.

---

2. Historical Origins and Motivations

The fundamental limitation addressed by attention mechanisms is the information bottleneck inherent to encoder-decoder recurrent neural networks (RNNs). In early sequence-to-sequence architectures (Sutskever et al., 2014), an encoder RNN compressed an entire input sequence into a single fixed-dimensional context vector, which the decoder then used to generate the output sequence. For long sequences, this compression proved severely detrimental, as the context vector could not reliably preserve fine-grained information from distant input positions.

2.1 The Bottleneck Problem in Sequence-to-Sequence Models

Cho et al. (2014) provided empirical evidence that the performance of encoder-decoder models degrades rapidly as sentence length increases, confirming the theoretical concern that a fixed-length vector is an insufficient intermediary for long-range dependency preservation. This observation motivated the search for mechanisms that would allow decoders to selectively access relevant portions of the encoder's hidden states at each decoding step.

2.2 Additive Attention: Bahdanau et al. (2015)

The seminal contribution of Bahdanau, Cho, and Bengio (2015) introduced what is now termed *additive attention* or *Bahdanau attention*. Rather than compressing the source sequence into a single vector, the model maintains all encoder hidden states and learns a differentiable alignment function

that scores the compatibility between each decoder hidden state and each encoder hidden state. Formally, the alignment score is computed as:

$$\text{score}(s_{t-1}, h_i) = v^T \tanh(W_s \cdot s_{t-1} + W_h \cdot h_i)$$

where s_{t-1} is the decoder hidden state, h_i is the i-th encoder hidden state, and v, W_s, W_h are learnable parameters. Scores are normalized via softmax to produce context-sensitive alignment weights, which are then used to compute a weighted sum of encoder states as the context vector for the current decoding step. This mechanism enabled the model to jointly learn to translate and align, yielding substantial improvements in neural machine translation.

2.3 Multiplicative Attention: Luong et al. (2015)

Luong, Pham, and Manning (2015) proposed a family of attention functions collectively termed *multiplicative* or *Luong attention*. The primary variant computes alignment via the inner product between query and key vectors:

$$\text{score}(q, k) = q^T \cdot k$$

A generalization inserts a learnable weight matrix between query and key:

$$\text{score}(q, k) = q^T W k$$

Luong attention is computationally more efficient than additive attention, particularly in higher-dimensional settings, because it avoids the tanh nonlinearity. Luong et al. also introduced the distinction between global attention, where all source positions are attended to, and local attention, where attention is restricted to a window around a predicted alignment point (Luong et al., 2015).

---

3. Mathematical Foundations of Attention Score Computation

Across all attention variants, the computation follows a unified three-stage structure: (1) projection of inputs into query, key, and value spaces; (2) computation of similarity scores between queries and keys; and (3) normalization and weighted aggregation over values. This section formalizes each stage.

3.1 Linear Projections

Given an input matrix $X \in \mathbb{R}^{n \times d}$, three learned projection matrices transform it into query (Q), key (K), and value (V) spaces:

$$Q = X W_Q, \quad K = X W_K, \quad V = X W_V$$

where $W_Q, W_K \in \mathbb{R}^{d \times d_k}$ and $W_V \in \mathbb{R}^{d \times d_v}$ are trainable parameter matrices. Each token thereby acquires a query vector (what it seeks), a key vector (what it offers), and a value vector (its actual content contribution).

3.2 Similarity Scoring

Multiple scoring functions have been proposed:

- **Dot-product:** score(i, j) = Q_i · K_j, efficient but magnitude-sensitive
- **Additive:** score(i, j) = v^T tanh(W_Q Q_i + W_K K_j), flexible but computationally heavier
- **General multiplicative:** score(i, j) = Q_i^T W K_j, parametric interpolation between the two

3.3 Scaled Dot-Product Attention

Vaswani et al. (2017) demonstrated that for large `d_k`, raw dot products grow in magnitude and push the softmax function into regions of near-zero gradient, destabilizing training. The solution is to scale by `√d_k` before normalization:

```
Attention(Q, K, V) = Softmax( Q K^T / √d_k ) · V
```

This scaling effectively controls the temperature of the softmax, preventing excessive sharpness. The result is a `n × d_v` output matrix where each row is a context-sensitive weighted combination of value vectors.

3.4 Geometric Interpretation

The dot product `Q_i · K_j = ||Q_i|| ||K_j|| cos(θ)` measures the cosine similarity between query and key vectors scaled by their norms. Attention is therefore a learned similarity metric in high-dimensional vector space. Following softmax normalization, the attention output is a convex combination of value vectors, placing the result on a probability simplex. Each output token representation is thus an adaptive interpolation across all value vectors, weighted by contextual relevance (Vaswani et al., 2017).

3.5 Masking in Score Computation

In autoregressive models, causal masking is applied by setting the attention score for future positions to $-\infty$ before softmax, ensuring that softmax assigns zero weight to those positions. This preserves the autoregressive generation property, which is fundamental in decoder-only models such as GPT-3 (Brown et al., 2020).

---

4. The Role of Softmax in Attention Mechanisms

The softmax function is the normalization operator at the heart of all standard attention mechanisms. It converts raw, unbounded attention scores (logits) into a valid probability distribution over positions, enabling the interpretation of attention weights as a probabilistic allocation of contextual focus.

4.1 Mathematical Definition

For a vector of scores `z = (z_1, z_2, ..., z_n)`, softmax produces:

```
Softmax(z_i) = exp(z_i) / Σ_{j=1}^{n} exp(z_j)
```

Key properties include: outputs bounded in (0, 1), all outputs summing to one, and disproportionate amplification of larger inputs through exponentiation. This last property endows softmax with a competitive selection character: tokens with marginally higher relevance scores receive substantially greater attention weight.

4.2 Necessity of Softmax in Attention

Normalization: Without softmax, raw attention scores lack a well-defined scale and cannot be interpreted as probabilistic weights over token positions.

Differentiability: Softmax is fully differentiable, enabling end-to-end gradient-based optimization through the attention computation (Bahdanau et al., 2015).

Competitive emphasis: Exponentiation amplifies score differences, ensuring that the model concentrates representation on the most contextually relevant positions.

4.3 Temperature and Softmax Sharpness

A temperature parameter `T` can modulate softmax sharpness: `Softmax(z_i / T)`. Values `T < 1` produce peakier distributions (harder attention), while `T > 1` produce flatter distributions (softer attention). In the Transformer, dividing by `√d_k` effectively acts as temperature scaling, preventing overly concentrated attention weights in high-dimensional spaces (Vaswani et al., 2017).

4.4 Limitations and Alternatives

Standard softmax attention carries inherent limitations. Its quadratic complexity $O(n^2)$ with sequence length makes application to very long documents computationally prohibitive. Additionally, softmax is sensitive to large logit values, potentially creating over-concentrated attention on a small subset of tokens, losing distributional information.

These limitations have motivated alternatives including sparsemax (Martins & Astudillo, 2016), which produces sparse probability distributions with exact zeros; kernel-based approximations as in the Performer (Choromanski et al., 2021); and structured sparse patterns as in the Longformer (Beltagy et al., 2020) and BigBird (Zaheer et al., 2020).

---

5. Self-Attention Mechanisms

Self-attention, also called intra-attention, is the mechanism whereby all queries, keys, and values within a single sequence are derived from the same input. This allows every position to directly attend to every other position, capturing long-range dependencies without the sequential bottleneck of recurrence.

5.1 Vanilla Self-Attention

In vanilla self-attention, the input X is projected to Q, K, V using the same matrix X as source. The resulting attention matrix $A \in \mathbb{R}^{n \times n}$ captures all pairwise relationships within the sequence. The complexity is $O(n^2 d)$, where n is the sequence length and d is the dimensionality (Vaswani et al., 2017). Self-attention enables global receptive fields without any locality prior, which was a decisive advantage over convolutional and recurrent alternatives.

5.2 Multi-Head Self-Attention

Vaswani et al. (2017) introduced multi-head attention as a mechanism for learning multiple relational subspaces in parallel. Rather than performing a single attention function, h independent attention heads each operate on distinct linear projections of the input:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\, W_O$$

$$\text{head}_i = \text{Attention}(Q\, W_Q^i,\ K\, W_K^i,\ V\, W_V^i)$$

where $W_Q^i, W_K^i \in \mathbb{R}^{d \times d_k}$ and $W_V^i \in \mathbb{R}^{d \times d_v}$, with $d_k = d_v = d/h$. Each head specializes in different relational patterns — syntactic dependencies, coreference, positional proximity — which are aggregated via the output projection W_O

5.3 Masked Self-Attention

Masked self-attention, employed in decoder-only models such as GPT-3 (Brown et al., 2020) and the original Transformer decoder (Vaswani et al., 2017), prevents each position from attending to subsequent positions. The causal mask ensures that the representation of each token depends only on prior context, preserving the autoregressive property necessary for language generation.

5.4 Bidirectional Self-Attention

In contrast to masked attention, bidirectional self-attention allows each token to attend to all other tokens without directional restriction. BERT (Devlin et al., 2019) exploits this to build deeply bidirectional contextual representations through masked language modeling, achieving strong performance on comprehension and classification tasks where full context is available at inference time.

---

6. The Transformer Architecture

The Transformer (Vaswani et al., 2017) constitutes the architectural watershed of modern AI. By eliminating recurrence and convolution entirely, it made attention the sole mechanism for both representation learning and sequence integration, enabling full parallelization during training and superior modeling of long-range dependencies.

6.1 Encoder and Decoder Stacks

The standard Transformer comprises stacked encoder and decoder blocks. Each encoder block contains a multi-head self-attention sublayer followed by a position-wise feed-forward network, with residual connections and layer normalization applied after each sublayer. The decoder block additionally contains a masked self-attention sublayer and a cross-attention sublayer that allows decoder positions to attend to encoder outputs.

6.2 Positional Encoding

Because self-attention is permutation-equivariant (it treats all positions identically), position must be explicitly injected. Vaswani et al. (2017) proposed sinusoidal positional encodings:

$$PE(pos, 2i) = \sin(pos / 10000^{2i/d})$$

$$PE(pos, 2i+1) = \cos(pos / 10000^{2i/d})$$

These encodings are added to the input embeddings and allow the model to leverage relative positional information. Subsequent work demonstrated that learned positional embeddings (Devlin et al., 2019) and relative positional biases (Dai et al., 2019) often yield further improvements.

6.3 Cross-Attention in Encoder-Decoder Models

Cross-attention is the mechanism by which decoder positions query encoder representations. Formally, queries originate from the decoder while keys and values come from the encoder output:

$$\text{CrossAttention}(Q_{dec}, K_{enc}, V_{enc}) = \text{Softmax}\left(Q_{dec} K_{enc}^T / \sqrt{d_k} \right) \cdot V_{enc}$$

This enables the decoder to selectively focus on relevant portions of the encoded source, which is critical in tasks such as machine translation, summarization, and image captioning. Cross-attention is also central to multimodal architectures including CLIP (Radford et al., 2021) and Stable Diffusion (Rombach et al., 2022).

6.4 Computational Complexity

The standard Transformer has $O(n^2 d)$ time and $O(n^2)$ space complexity due to the full attention matrix computation. For sequences of length n, this quadratic scaling becomes prohibitive beyond a few thousand tokens, which motivated the development of efficient attention variants (Section 9).

Component	Function	Complexity
Self-Attention	Intra-sequence dependency modeling	$O(n^2 d)$
Cross-Attention	Inter-sequence conditional reasoning	$O(n \cdot m \cdot d)$
Multi-Head Attention	Parallel relational subspace learning	$O(h \cdot n^2 \cdot d/h)$

Component	Function	Complexity
Feed-Forward Network	Position-wise nonlinear transformation	O(n·d·d_ff)
Positional Encoding	Sequence order injection	O(n·d)

7. Mechanistic Interpretability of Attention Heads

A central question in transformer research extends beyond what attention mechanisms do computationally to what differentiated relational structures each head internalizes during training. Large-scale interpretability studies on models including BERT (Devlin et al., 2019) and GPT-3 (Brown et al., 2020) have revealed recurring functional motifs across attention heads, demonstrating that, while knowledge is distributed across layers and parameters, individual heads exhibit measurable specialization in the relational patterns they amplify.

7.1 Formal Function of an Individual Attention Head

Each attention head in a multi-head layer contains three independently learned linear transformations: a query matrix `W_Q`, a key matrix `W_K`, and a value matrix `W_V`. For every token representation `x`, the head computes:

```
Q = x W_Q,    K = x W_K,    V = x W_V
```

Attention weights are derived via the scaled dot-product of `Q` and `K`, followed by softmax normalization, and the head output is a weighted sum of `V` vectors. Because each head possesses its own distinct projection matrices `W_Q, W_K, W_V`, it learns a unique relational mapping across tokens — a distinct geometric lens on the embedding space (Vaswani et al., 2017). Crucially, attention heads do not store symbolic rules; they learn parameterized projections that amplify particular relational structures statistically present in training data.

7.2 Syntactic Specialization

Systematic attention visualization and probing experiments have demonstrated that certain heads consistently encode grammatical dependencies. Clark et al. (2019) found that specific BERT attention heads align strongly with dependency parse relations, including subject-verb agreement, direct object linking, determiner-noun attachment, and prepositional phrase attachment. In the sentence *"The keys to the cabinet are missing,"* designated heads strongly connect the head noun *keys* to the verb *are*, reflecting learned number agreement tracking. Tenney et al. (2019) extended this analysis, demonstrating that BERT effectively rediscovers the classical NLP pipeline across its layers: lower layers capture surface features, middle layers capture syntactic structure, and upper layers capture semantic relations — an emergent hierarchy not explicitly designed.

7.3 Positional and Structural Patterns

Beyond semantic content, a subset of heads learns positional biases largely independent of token identity. Voita et al. (2019) identified a class of positional heads in machine translation models that attend primarily to adjacent tokens — most commonly the immediately preceding token. These heads encode structural scaffolding: sentence boundary detection, paragraph transition signaling, and, in code-specialized models, indentation and block structure tracking. Notably, Voita et al. (2019) demonstrated through head pruning experiments that these positional heads, along with syntactic heads, constitute the functionally critical minority; the majority of heads can be pruned without measurable performance degradation.

7.4 Coreference and Entity Tracking

Mid-to-upper layers of transformer models host heads that specialize in coreference resolution — the linking of pronouns and definite noun phrases to their referential antecedents. Clark et al. (2019) documented heads that consistently connect pronouns such as "she" or "they" to their nominal antecedents across considerable spans of intervening text. This behavior is particularly prominent in models pretrained on corpora rich in narrative discourse. The emergence of coreference-sensitive heads in mid-layers, rather than lower layers, is consistent with the hypothesis that entity tracking requires first building a syntactic parse and then reasoning about discourse structure (Tenney et al., 2019).

7.5 Semantic Role and Thematic Relation Heads

In the upper layers of deep transformer stacks, attention heads capture abstract semantic relations that extend beyond surface syntax. These include agent-action associations, action-patient relations, modifier-modified concept links, and cause-effect relationships. Vig and Belinkov (2019) provided fine-grained visualization evidence that upper-layer heads preferentially encode thematic role structure, with the degree of specialization correlating with model depth and training corpus size. These semantic heads form the representational substrate upon which task-specific fine-tuning operates.

7.6 Induction Heads and In-Context Learning

A particularly well-documented phenomenon in autoregressive transformers is the *induction head* (Olsson et al., 2022). Induction heads detect repeated token sub-sequences and copy continuation patterns: if a sequence [A, B, ..., A] appears in context, an induction head attending to the second occurrence of A will assign high weight to the token B that followed the first occurrence, effectively implementing a copy-and-complete mechanism. Olsson et al. (2022) demonstrated that induction heads form a two-head circuit: a *previous token head* that attends to the token immediately preceding any given position, and an *induction head proper* that attends to the position immediately following the previous occurrence of the current token. This circuit, which forms reliably during a distinct phase

transition in training, is considered a primary mechanistic substrate for in-context learning — the capacity of large language models to adapt to novel tasks from a few examples without gradient updates.

7.7 Distributed Versus Specialized Learning

Although interpretability research frequently identifies apparently specialized heads, three important qualifications apply. First, knowledge is distributed across many heads and layers; ablating a single head rarely destroys a capability entirely. Second, multiple heads frequently encode redundant representations of the same structural pattern (Michel et al., 2019). Third, head function is not entirely fixed: the same head may contribute to different computations depending on input context. Michel et al. (2019) systematically demonstrated that the vast majority of attention heads in large models can be pruned at test time with minimal performance loss, confirming that specialization operates within a highly redundant distributed system. This redundancy may confer robustness to noise and partial corruption of the model's parameters.

7.8 Layerwise Functional Differentiation

Empirical studies converge on a rough functional hierarchy across transformer depth. Tenney et al. (2019) used probing classifiers to quantify where different linguistic properties are best encoded, finding consistent layerwise ordering:

Layer Range	Dominant Functional Motifs	Representative Head Types
Early (1-4)	Local syntax, word identity, positional dependencies	Positional, n-gram, surface pattern heads
Middle (5-8)	Grammatical structure, dependency arcs, coreference	Syntactic, entity tracking, agreement heads
Upper (9-12+)	Abstract semantics, thematic roles, task reasoning	Semantic role, induction, long-range heads

This layered progression mirrors increasing abstraction from surface form toward conceptual meaning, though it emerges from gradient descent rather than any explicit architectural constraint (Tenney et al., 2019; Clark et al., 2019).

7.9 Theoretical Interpretation

Conceptually, each attention head learns a projection subspace optimized to minimize prediction error on its training objective — typically next-token cross-entropy loss for language models, or masked token prediction loss for BERT-style models (Devlin et al., 2019). The head's emergent behavior arises entirely from gradient descent shaping its parameters across vast corpora to reduce that loss. Attention heads therefore do not *"understand"* grammar or semantics in any symbolic sense;

they internalize statistical regularities that approximate linguistic structure because doing so is the most efficient path to reducing prediction error on naturalistic text (Voita et al., 2019). The convergence of independently trained models on similar head specialization patterns — syntactic heads, positional heads, induction heads — constitutes evidence that these structural categories reflect genuine properties of natural language statistical structure, not artifacts of any particular training procedure.

---

9. Training Attention Models: Backpropagation and Optimization

Attention mechanisms are differentiable functions fully integrated into the computational graph, and their parameters are learned end-to-end via gradient-based optimization. This section details the training dynamics specific to attention-based architectures.

7.1 The Training Pipeline

Training proceeds through four stages: (1) forward pass, wherein inputs are processed through embeddings, attention layers, and output projections to produce predictions; (2) loss computation, quantifying discrepancy between predictions and targets; (3) backward pass via backpropagation, computing gradients of the loss with respect to all parameters; and (4) parameter update via an optimizer.

7.2 Loss Functions

For language modeling, the standard loss is cross-entropy over the vocabulary:

$$L = - \Sigma_t \log P(w_t \mid w_{<t})$$

For classification and understanding tasks such as those addressed by BERT (Devlin et al., 2019), masked language modeling and next-sentence prediction objectives are employed during pretraining. For regression tasks, mean squared error is standard.

7.3 Gradient Flow Through Attention

Backpropagation through scaled dot-product attention applies the chain rule through the softmax, matrix multiplications, and the scaling operation. Crucially, all operations are differentiable, enabling gradients to flow from the loss through value aggregation, through softmax normalization, through the similarity computation, and back to the query, key, and value projection matrices.

The residual connections present in each Transformer block (He et al., 2016) and layer normalization (Ba et al., 2016) are critical to stable gradient flow through deep stacks. Without these, attention-based models of 12 or more layers would suffer from vanishing or exploding gradients.

7.4 Optimization Algorithms

The Adam optimizer (Kingma & Ba, 2015) is the dominant choice for training Transformer models, combining momentum and adaptive learning rates. AdamW (Loshchilov & Hutter, 2019), which decouples weight decay from the adaptive update, is preferred for large pretrained models including BERT and GPT variants. The standard learning rate schedule employs linear warmup followed by decay, as originally specified by Vaswani et al. (2017):

```
lr = d_model^{-0.5} · min(step^{-0.5}, step · warmup_steps^{-1.5})
```

7.5 No Internal Error Correction in Attention

It is important to clarify that attention mechanisms contain no intrinsic error-correction mechanism. Attention is a forward-pass computation: given inputs, it produces weighted aggregations. Error detection and correction occur exclusively outside attention, through the loss function and backpropagation. Attention parameters are updated indirectly as part of the global gradient descent procedure. During inference, after training is complete, no parameter updates occur; the model performs only a forward pass.

---

10. Transformer-Derived Language Models

The Transformer's architectural flexibility enabled a suite of highly successful language models, differentiated primarily by whether they employ encoder-only, decoder-only, or encoder-decoder configurations.

8.1 Encoder-Only Models: BERT

BERT (Devlin et al., 2019) uses a stack of bidirectional self-attention encoder layers, pretrained via masked language modeling (MLM) and next-sentence prediction on large corpora. BERT's bidirectional context modeling makes it particularly effective for natural language understanding tasks: classification, named entity recognition, question answering, and semantic similarity. Subsequent variants including RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2020) refined the pretraining procedure and parameter efficiency.

8.2 Decoder-Only Models: GPT-3

GPT-3 (Brown et al., 2020) employs a decoder-only Transformer with masked (causal) self-attention, pretrained autoregressively on 300 billion tokens. With 175 billion parameters, GPT-3 demonstrated remarkable few-shot learning capabilities, generating coherent text, completing code, and answering questions without task-specific fine-tuning. Its success established the scaling law paradigm: model capability scales predictably with model size, data volume, and compute (Kaplan et al., 2020).

8.3 Encoder-Decoder Models: T5

T5 (Raffel et al., 2020) frames all NLP tasks as text-to-text problems, mapping input text to output text regardless of task type. It employs a full Transformer encoder-decoder with both bidirectional encoding and autoregressive decoding. A systematic study of pretraining objectives, architectures, and data mixtures at scale, T5 provides a unified framework that achieves strong performance across translation, summarization, classification, and question answering.

Model	Architecture	Attention Type	Pretraining Objective	Primary Application
BERT	Encoder-only	Bidirectional Self-Attention	Masked LM + NSP	NLU tasks
GPT-3	Decoder-only	Masked Self-Attention	Autoregressive LM	Generation, few-shot
T5	Encoder-Decoder	Bidirectional Enc + Causal Dec	Text-to-text denoising	Multi-task NLP
Transformer-XL	Decoder-only + Recurrence	Relative Pos. Attention	Autoregressive LM	Long-context modeling
BERT-XL/RoBERTa	Encoder-only	Bidirectional Self-Attention	MLM only	Robust NLU

11. Efficient and Sparse Attention Variants

The quadratic $O(n^2)$ complexity of full self-attention renders it computationally intractable for sequences exceeding a few thousand tokens. A substantial body of research has developed attention variants that reduce this cost while preserving, to varying degrees, the modeling capabilities of full attention.

9.1 Sparse Attention (Sparse Transformer)

Child et al. (2019) introduced sparse attention patterns that restrict each token to attending to only a subset of positions. Two structured patterns are proposed: strided attention, wherein each token attends to every r-th position, and fixed attention, wherein certain designated summary positions attend globally. The resulting complexity is $O(n\sqrt{n})$, yielding substantial savings for long sequences.

9.2 Local Attention

Local attention restricts each token to attending within a sliding window of fixed radius w, reducing complexity to $O(n \cdot w)$. While this loses global context, local attention is sufficient for many tasks with naturally local dependencies and is combined with global attention in more powerful models.

9.3 Longformer

Beltagy et al. (2020) proposed the Longformer, which combines local windowed attention with global attention on a small set of task-relevant tokens (e.g., the [CLS] token). This combination achieves linear O(n) complexity while preserving the ability to propagate global context. Longformer demonstrated strong performance on long-document classification and question answering tasks, substantially outperforming prior methods on the SCROLLS benchmark.

9.4 Reformer

Kitaev et al. (2020) addressed memory constraints via locality-sensitive hashing (LSH) attention. Rather than computing all n^2 pairs, Reformer uses LSH to identify approximately similar query-key pairs and restricts attention computation to these pairs. Combined with reversible residual layers that eliminate the need to store all intermediate activations, Reformer achieves O(n log n) complexity and substantially reduced memory usage, enabling Transformer modeling on sequences of 64,000 tokens or more.

9.5 BigBird

Zaheer et al. (2020) proposed BigBird, which combines random attention (each token attends to a random subset of positions), local windowed attention, and global attention on a set of special tokens. This combination is theoretically equivalent to a universal Turing machine and guarantees approximation of any function of the sequence, while achieving O(n) complexity. BigBird achieves state-of-the-art results on genomics tasks requiring processing of very long DNA sequences.

9.6 Linformer

Wang et al. (2020) demonstrated that the attention matrix has low stable rank in practice and proposed the Linformer, which projects keys and values to a lower-dimensional space of size k (independent of n) before attention computation. This yields O(n·k) complexity, linear in sequence length, at a modest approximation cost.

9.7 Performer

Choromanski et al. (2021) developed the Performer using Fast Attention Via positive Orthogonal Random features (FAVOR+), a kernel approximation technique that decomposes the softmax attention kernel into a product of random feature maps, enabling linear-complexity attention without approximating the attention matrix directly. Performer can approximate standard Transformer attention with theoretical guarantees while achieving O(n) time and space complexity.

9.8 Flash Attention

Dao et al. (2022) introduced FlashAttention, which reframes attention as an IO-aware algorithm optimized for modern GPU memory hierarchies. Rather than materializing the full n×n attention matrix in high-bandwidth memory (HBM), FlashAttention tiles the computation and exploits fast on-chip

SRAM. The algorithm is mathematically equivalent to standard attention (no approximation) but requires substantially fewer memory reads/writes, achieving 2–4× wall-clock speedup in practice and enabling training on sequences up to 16,384 tokens within standard GPU memory constraints.

Variant	Core Mechanism	Complexity	Key Limitation
Full Attention	Dense pairwise scoring	$O(n^2)$	Quadratic scaling
Sparse Transformer	Structured sparse patterns	$O(n\sqrt{n})$	Limited global context
Longformer	Local + global attention	$O(n)$	Fixed global token count
Reformer	LSH attention + reversible layers	$O(n \log n)$	Hash collision sensitivity
BigBird	Random + local + global	$O(n)$	Approximation trade-offs
Linformer	Low-rank projection of K, V	$O(n \cdot k)$	Information compression
Performer	Kernel (FAVOR+) approximation	$O(n)$	Approximation error
FlashAttention	IO-aware exact attention	$O(n^2)$ ops, $O(n)$ HBM	Requires GPU SRAM tiling

12. Attention Mechanisms in Computer Vision

Attention mechanisms were adapted to visual domains initially as auxiliary modules within convolutional architectures, and subsequently as standalone architectures replacing convolution entirely.

10.1 Spatial and Channel Attention

Spatial attention assigns scalar weights to different spatial positions in a feature map, enabling the network to focus on task-relevant image regions. Channel attention assigns weights to feature channels, recalibrating the relative importance of different learned features. The Squeeze-and-Excitation Networks (Hu et al., 2018) demonstrated that channel attention provides consistent improvements when integrated into convolutional architectures, suggesting that cross-channel dependencies carry significant representational value.

10.2 Vision Transformer (ViT)

Dosovitskiy et al. (2021) demonstrated that a pure Transformer, applied directly to sequences of image patches, achieves excellent performance on image classification. An image of resolution $H \times W \times C$ is divided into $N = HW/P^2$ non-overlapping patches of size $P \times P$. Each patch is linearly

projected to a d-dimensional embedding, and a learnable [CLS] token is prepended. Standard Transformer encoder self-attention is then applied to this sequence. ViT matches or exceeds convolutional networks when pretrained on sufficiently large datasets (ImageNet-21K or JFT-300M), demonstrating that inductive biases of convolution (locality, translation equivariance) are not strictly necessary for vision at scale.

10.3 Detection Transformers (DETR)

Carion et al. (2020) introduced DETR (DEtection TRansformer), which reformulates object detection as a direct set prediction problem. A CNN backbone extracts features, which are flattened and processed by a Transformer encoder. The decoder attends to encoder outputs via cross-attention, with a fixed set of learned object queries, producing a fixed-size set of predictions. DETR eliminates hand-crafted post-processing stages such as non-maximum suppression and anchor generation, providing a cleaner and more principled detection pipeline.

10.4 Axial Attention

Axial attention (Ho et al., 2019) factorizes 2D self-attention along each axis independently. Instead of computing attention over all HW positions simultaneously (quadratic in image area), axial attention applies one-dimensional attention along rows and columns separately, reducing complexity from $O((HW)^2)$ to $O(HW(H+W))$ while preserving global receptive fields within each axial pass.

---

13. Multimodal Attention Models

Cross-attention is the primary mechanism for integrating information across modalities in multimodal AI systems.

11.1 CLIP

Radford et al. (2021) introduced CLIP (Contrastive Language-Image Pretraining), which learns a shared embedding space for images and text via contrastive learning on 400 million image-text pairs. A Transformer-based text encoder and a vision encoder (either ResNet or ViT) produce embeddings that are aligned through cross-modal contrastive loss. CLIP demonstrates remarkable zero-shot classification capability, transferring to novel tasks by computing cosine similarity between image embeddings and text embeddings of candidate class names.

11.2 Stable Diffusion and Diffusion Transformers

Rombach et al. (2022) integrated cross-attention into latent diffusion models, enabling fine-grained text-conditioned image generation. At each denoising step, the UNet decoder attends to text encoder outputs via cross-attention:

```
CrossAttention(Q_img, K_text, V_text) = Softmax( Q_img K_text^T / √d_k )
                                        · V_text
```

This mechanism allows the model to progressively inject textual semantics into the generated image at multiple scales and denoising timesteps, producing high-fidelity, compositionally complex images conditioned on natural language descriptions.

---

14. Graph Attention Networks

Graph-structured data presents unique challenges for attention mechanisms, as the notion of sequence order does not apply and connectivity is defined by graph topology rather than position.

12.1 Graph Attention Networks (GAT)

Velickovic et al. (2018) introduced Graph Attention Networks (GATs), which apply masked self-attention to graph neighborhoods. For each node, attention coefficients are computed between the node and its neighbors, enabling the model to assign differentially weighted contributions to neighboring nodes during aggregation:

```
α_{ij} = Softmax_j( LeakyReLU( a^T [W h_i || W h_j] ) )
```

where `h_i, h_j` are node feature vectors, `W` is a shared linear transformation, and `a` is an attention vector. The use of multi-head attention in GATs provides implicit regularization and richer node representations. GATs achieve state-of-the-art performance on node classification benchmarks and generalize to graphs with varying topology without modification.

---

15. Memory-Augmented and Hierarchical Attention

13.1 Memory Networks

Sukhbaatar et al. (2015) proposed end-to-end memory networks, which augment attention-based sequence models with an external memory matrix. The model reads from memory via attention-weighted sums and can perform multiple read-write cycles (memory hops). This enables explicit storage and retrieval of factual information, addressing limitations of purely parametric memory in standard architectures.

13.2 Transformer-XL and Relative Positional Attention

Transformer-XL (Dai et al., 2019) addresses the fixed-length context limitation of standard Transformers by introducing segment-level recurrence. Hidden states from previous segments are cached and used as an extended context for the current segment, enabling effective context windows that span multiple times the length of a single segment. Crucially, Transformer-XL introduces a

relative positional encoding scheme that correctly handles the positionally heterogeneous extended context, resolving inconsistencies that would arise from applying absolute positional encodings across segment boundaries.

13.3 Hierarchical Attention

Hierarchical attention (Yang et al., 2016) applies attention at multiple levels of granularity. In document classification, word-level attention aggregates word representations into sentence vectors, and sentence-level attention aggregates sentence vectors into a document representation. Each level focuses on the most informative units at that granularity, mirroring the hierarchical structure of natural language. Hierarchical attention models have been applied to long-document understanding, sentiment analysis, and multi-document summarization.

---

16. Hard Attention and Soft Attention in Reinforcement Learning

14.1 Soft Attention

Soft attention computes a continuous probability distribution over all positions, producing attention weights that are smooth, differentiable functions of the input. This allows end-to-end gradient-based training and is the approach adopted by all major Transformer architectures. Soft attention aggregates all positions, weighted by relevance, and is therefore inherently global.

14.2 Hard Attention

Hard attention selects a discrete subset of input positions via stochastic sampling, rather than computing a weighted average over all positions. Because sampling is non-differentiable, hard attention models must be trained using reinforcement learning (specifically, REINFORCE; Williams, 1992) or other gradient estimation techniques. Mnih et al. (2014) demonstrated hard attention in recurrent visual attention models for image classification, showing that models can learn to sequentially focus on informative image regions. While hard attention produces more interpretable, sparser alignments, the variance of gradient estimates during training is typically higher than with soft attention, and convergence is generally slower.

---

17. Comprehensive Taxonomy and Classification

The following tables synthesize the full taxonomy of attention mechanisms discussed in this survey.

15.1 Core Mechanisms by Structure

Mechanism	Score Function	Q Source	K/V Source	Complexity
Additive (Bahdanau)	v^T tanh(W_Q q + W_K k)	Decoder state	Encoder states	O(n·m)
Multiplicative (Luong)	q^T k or q^T W k	Decoder state	Encoder states	O(n·m)
Scaled Dot-Product	Q K^T / √d_k	Same sequence	Same sequence	O(n²d)
Multi-Head Attention	h × Scaled Dot-Product	Same or cross	Same or cross	O(n²d)
Masked Self-Attention	Causal masked scaled DP	Same sequence	Past only	O(n²d)
Cross-Attention	Scaled Dot-Product	Target seq	Source seq	O(n·m·d)
Graph Attention (GAT)	LeakyReLU + softmax	Node features	Neighbor features	O(|E|·d)

15.2 Conceptual Classification

Category	Core Idea	Representative Model	Use Case
Additive Attention	Neural scoring function	Bahdanau et al. (2015)	Neural Machine Translation
Dot-Product Attention	Vector inner product similarity	Luong et al. (2015)	Seq2Seq translation
Scaled Dot-Product	Temperature-controlled inner product	Vaswani et al. (2017)	All Transformer tasks
Multi-Head Self-Attention	Parallel relational subspaces	BERT, GPT-3	Language modeling, NLU
Sparse/Efficient Attention	Structured or approximate attention	Longformer, BigBird	Long document tasks
Vision Attention	Patch-level global self-attention	ViT (Dosovitskiy et al., 2021)	Image classification
Cross-Modal Attention	Inter-modality information fusion	CLIP, Stable Diffusion	Vision-language tasks
Graph Attention	Neighborhood-weighted aggregation	GAT (Velickovic et al., 2018)	Graph classification, NER
Memory Attention	External memory read/write	Memory Networks, Transformer-XL	Long-context reasoning

18. Discussion and Concluding Perspective

Attention mechanisms represent a fundamental epistemic shift in machine learning: from compressive, fixed-length representations toward dynamically contextual relational computation. The progression from additive attention in sequence-to-sequence models (Bahdanau et al., 2015) through the fully attention-based Transformer (Vaswani et al., 2017) to large-scale pretrained models (Devlin et al., 2019; Brown et al., 2020) and vision transformers (Dosovitskiy et al., 2021) reflects a consistent trend toward architectures that reason relationally at scale.

Three core principles unify the diversity of attention variants surveyed here. First, attention operationalizes relevance: it converts similarity into a differentiable weighting of information. Second, attention is compositional: it can be stacked, parallelized into multiple heads, restricted to sparse patterns, or extended across modalities and graph topologies without changing its fundamental operation. Third, attention is trained: all parameters of all attention mechanisms discussed herein are learned end-to-end via gradient descent, and no component performs error correction independently of the global optimization process.

The primary open challenges concern efficiency at scale, interpretability of learned attention patterns, and the integration of attention-based reasoning with structured knowledge. Flash Attention (Dao et al., 2022) has addressed IO efficiency substantially, and sparse methods (Beltagy et al., 2020; Zaheer et al., 2020) extend the sequence length frontier, but the fundamental $O(n^2)$ computational structure remains a constraint. Interpretability methods that attribute model behavior to specific attention heads remain an active research area (Jain & Wallace, 2019; Wiegreffe & Pinter, 2019). And the synthesis of attention with symbolic, graph-based, and retrieval-augmented reasoning constitutes a promising frontier for next-generation AI systems.

As large language models based on Transformer attention continue to scale — with models such as GPT-4, Gemini, and Claude demonstrating emergent capabilities not observed at smaller scales (Wei et al., 2022) — the rigorous understanding of attention mechanisms surveyed here becomes increasingly important for researchers and practitioners designing, evaluating, and deploying these systems responsibly.

References

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. arXiv preprint arXiv:1607.06450. https://arxiv.org/abs/1607.06450

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. Proceedings of the International Conference on Learning Representations (ICLR 2015). https://arxiv.org/abs/1409.0473

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150. https://arxiv.org/abs/2004.05150

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., & others. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877-1901. https://arxiv.org/abs/2005.14165

Carion, N., Massa, F., Synnaert, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision (ECCV 2020) (pp. 213-229). Springer. https://arxiv.org/abs/2005.12872

Child, R., Gray, S., Radford, A., & Sutskever, I. (2019). Generating long sequences with sparse transformers. arXiv preprint arXiv:1904.10509. https://arxiv.org/abs/1904.10509

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014) (pp. 1724-1734). ACL. https://arxiv.org/abs/1406.1078

Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., & others. (2021). Rethinking attention with performers. Proceedings of the International Conference on Learning Representations (ICLR 2021). https://arxiv.org/abs/2009.14794

Clark, K., Khandelwal, U., Levy, O., & Manning, C. D. (2019). What does BERT look at? An analysis of BERT's attention. In Proceedings of the 2019 BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP (pp. 276-286). ACL. https://arxiv.org/abs/1906.04341

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., & Salakhutdinov, R. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019) (pp. 2978-2988). ACL. https://arxiv.org/abs/1901.02860

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., & Ré, C. (2022). FlashAttention: Fast and memory-efficient exact attention with IO-awareness. Advances in Neural Information Processing Systems, 35, 16344-16359. https://arxiv.org/abs/2205.14135

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2019) (pp. 4171-4186). ACL. https://arxiv.org/abs/1810.04805

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., & others. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. Proceedings of the International Conference on Learning Representations (ICLR 2021). https://arxiv.org/abs/2010.11929

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016) (pp. 770-778). IEEE. https://arxiv.org/abs/1512.03385

Ho, J., Kalchbrenner, N., Weissenborn, D., & Salimans, T. (2019). Axial attention in multidimensional transformers. arXiv preprint arXiv:1912.12180. https://arxiv.org/abs/1912.12180

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018) (pp. 7132-7141). IEEE. https://arxiv.org/abs/1709.01507

Jain, S., & Wallace, B. C. (2019). Attention is not explanation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2019) (pp. 3543-3556). ACL. https://arxiv.org/abs/1902.10186

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). Scaling laws for neural language models. arXiv preprint arXiv:2001.08361. https://arxiv.org/abs/2001.08361

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. Proceedings of the International Conference on Learning Representations (ICLR 2015). https://arxiv.org/abs/1412.6980

Kitaev, N., Kaiser, L., & Levskaya, A. (2020). Reformer: The efficient transformer. Proceedings of the International Conference on Learning Representations (ICLR 2020). https://arxiv.org/abs/2001.04451

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. Proceedings of the International Conference on Learning Representations (ICLR 2020). https://arxiv.org/abs/1909.11942

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692. https://arxiv.org/abs/1907.11692

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. Proceedings of the International Conference on Learning Representations (ICLR 2019). https://arxiv.org/abs/1711.05101

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2015) (pp. 1412-1421). ACL. https://arxiv.org/abs/1508.04025

Martins, A., & Astudillo, R. (2016). From softmax to sparsemax: A sparse model of attention and multi-label classification. In Proceedings of the International Conference on Machine Learning (ICML 2016) (pp. 1614-1623). PMLR. https://arxiv.org/abs/1602.02068

Michel, P., Levy, O., & Neubig, G. (2019). Are sixteen heads really better than one? Advances in Neural Information Processing Systems, 32. https://arxiv.org/abs/1905.10650

Mnih, V., Heess, N., Graves, A., & Kavukcuoglu, K. (2014). Recurrent models of visual attention. Advances in Neural Information Processing Systems, 27. https://arxiv.org/abs/1406.6247

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Johnston, S., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., & Olah, C. (2022). In-context learning and induction heads. Transformer Circuits Thread. https://arxiv.org/abs/2209.11895

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., & others. (2021). Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning (ICML 2021) (pp. 8748-8763). PMLR. https://arxiv.org/abs/2103.00020

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140), 1-67. https://jmlr.org/papers/v21/20-074.html

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022) (pp. 10684-10695). IEEE. https://arxiv.org/abs/2112.10752

Sukhbaatar, S., Szlam, A., Weston, J., & Fergus, R. (2015). End-to-end memory networks. Advances in Neural Information Processing Systems, 28. https://arxiv.org/abs/1503.08895

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. Advances in Neural Information Processing Systems, 27. https://arxiv.org/abs/1409.3215

Tenney, I., Das, D., & Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019) (pp. 4593-4601). ACL. https://arxiv.org/abs/1905.05950

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30. https://arxiv.org/abs/1706.03762

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2018). Graph attention networks. Proceedings of the International Conference on Learning Representations (ICLR 2018). https://arxiv.org/abs/1710.10903

Vig, J., & Belinkov, Y. (2019). Analyzing the structure of attention in a transformer language model. In Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP (pp. 63-76). ACL. https://arxiv.org/abs/1906.04284

Voita, E., Talbot, D., Moiseev, F., Sennrich, R., & Titov, I. (2019). 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 (ACL 2019) (pp. 5797-5808). ACL. https://arxiv.org/abs/1905.09418

Wang, S., Li, B. Z., Khabsa, M., Fang, H., & Ma, H. (2020). Linformer: Self-attention with linear complexity. arXiv preprint arXiv:2006.04768. https://arxiv.org/abs/2006.04768

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., & others. (2022). Emergent abilities of large language models. Transactions on Machine Learning Research. https://arxiv.org/abs/2206.07682

Wiegreffe, S., & Pinter, Y. (2019). Attention is not not explanation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2019) (pp. 11-20). ACL. https://arxiv.org/abs/1908.04626

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8(3-4), 229-256. https://doi.org/10.1007/BF00992696

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2016) (pp. 1480-1489). ACL. https://doi.org/10.18653/v1/N16-1174

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., & Ahmed, A. (2020). Big bird: Transformers for longer sequences. Advances in Neural Information Processing Systems, 33, 17283-17297. https://arxiv.org/abs/2007.14062