Iqra Journal of Engineering & Computing
Vol. 02, No. 01, (March 2026) 74 –97
DOI 10.21621/ijec.20260201.05

A Comparative Review of Transformer Architectures: Evolution, Efficiency, Applications, and Future Directions

Israr Ali[1*], Ali Ahmed Siddiqui [1], Aarij Mahmood Hussaan [1]

ABSTRACT

Transformer architectures have become a central foundation of modern artificial intelligence because of their scalability, parallel computation, and representation learning across language, vision, code, and multimodal domains. Since the original Transformer, the architecture has evolved into encoder-only, decoder-only, encoder-decoder, efficient attention, vision, multimodal, sparse Mixture-of-Experts, open-weight, state-space, recurrent, and hybrid model families. This review presents a structured and benchmark-informed comparison of Transformer architectures using a defined literature selection protocol, inclusion and exclusion criteria, model-family coding dimensions, and source-verification rules. The paper compares major architectures with respect to training objective, attention or sequence-modeling mechanism, computational complexity, scalability, context length, modality support, openness, benchmark behavior, reproducibility, deployment suitability, and architectural trade-offs. Representative benchmarks, including GLUE/SuperGLUE, MMLU/MMLU-Pro, HELM, HumanEval, SWE-bench, LongBench, RULER, ImageNet, and MMMU, are used to support critical discussion rather than relying only on descriptive model summaries. Recent developments, including GPT-4o, Gemini 2.5, Llama 4, DeepSeek-R1, Qwen3, Jamba, RecurrentGemma/Griffin, Hyena, Mamba, and xLSTM, are incorporated to reflect the 2024-2025 research landscape. The review identifies evidence-backed gaps in long-context reasoning, multimodal evaluation, cost-aware comparison, open versus closed model reproducibility, safety, interpretability, and domain-specific adaptation. Overall, the study provides a rigorous comparative foundation for researchers and practitioners evaluating Transformer-based and hybrid artificial intelligence systems.

Keywords: Transformer, self-attention, BERT, GPT, large language models, multimodal learning, Mixture-of-Experts, efficient attention, Vision Transformer, state-space models, benchmark evaluation, long-context reasoning, open-weight models

Author's Affiliation:
Institution(s) Name: [1] Iqra University Main Campus, Karachi,

Country: [1] Pakistan,
Corresponding Author's Email: [1]Israr.ali@iqra.edu.pk

3105-4528 (Online), 3105-451X (Print) © 2025

1. Introduction

Transformer architectures have become one of the most influential developments in modern artificial intelligence due to their ability to support parallel computation, model long-range dependencies, and scale effectively with large datasets and computational resources. Introduced by Vaswani in 2017, the original Transformer replaced recurrent and convolutional sequence modeling with a self-attention mechanism that allows tokens in a sequence to interact directly with one another [1]. This innovation significantly improved sequence modeling and later became the foundation for many advanced models in natural language processing, computer vision, speech processing, software engineering, code intelligence, and multimodal generative artificial intelligence.

Before the development of Transformers, sequence learning was largely dominated by recurrent neural networks, long short-term memory networks, gated recurrent units, and convolutional neural networks. Although these architectures achieved useful results, they faced several limitations, including sequential computation, difficulty in capturing long-distance dependencies, vanishing gradients, and limited scalability for large-scale training. The self-attention mechanism addressed many of these limitations by enabling direct relationships among all tokens in a sequence and allowing efficient parallel processing during training.

Since the introduction of the original Transformer, the field has expanded into several major architectural families. Encoder-only models such as BERT and its variants improved language understanding through bidirectional contextual representation and masked language modeling [2]–[6]. Decoder-only models such as GPT, GPT-2, GPT-3, PaLM, LLaMA, Qwen, and DeepSeek advanced generative artificial intelligence through autoregressive pre-training, in-context learning, instruction following, and large-scale reasoning capabilities [7],[11], [25],[29]. Encoder-decoder models such as T5 and BART provided strong performance for sequence-to-sequence tasks, including summarization, translation, and text transformation [12], [13].

In addition to language-based models, Transformer research has expanded toward efficient long-context processing, computer vision, multimodal learning, and sparse expert-based scaling. Efficient models such as Linformer, Performer, Longformer, BigBird, and FlashAttention attempt to reduce the computational and memory limitations of standard self-attention [14]–[18]. Vision Transformer and Swin Transformer extended Transformer architectures to image classification, object detection, and visual representation learning [19], [20]. Multimodal models such as CLIP, GPT-4, and Gemini demonstrated the ability of Transformer-based systems to connect text, image, audio, video, and other modalities [21], [22]. Sparse Mixture-of-Experts models such as Switch Transformer, GShard, Mixtral, DeepSeek-V3, and Qwen3 further improved scalability by activating only selected expert networks for each input [23],[27].

Despite their success, Transformer architectures continue to face several challenges. Standard self-attention has quadratic complexity with respect to sequence length, making long-context processing computationally expensive. Large-scale Transformer models also require extensive training data, high computational resources, and significant energy consumption. Furthermore, generative Transformer models may produce hallucinated

information, exhibit bias, lack interpretability, and create safety and reproducibility concerns. Recent alternatives such as RWKV, RetNet, Mamba, and Mamba-2 have been proposed to address some of these limitations by introducing recurrent-style, retention-based, and state-space modelling approaches [30], [33].

This paper presents a structured comparative literature review of Transformer architectures from the original Transformer to modern foundation models and emerging alternatives. Unlike narrow reviews that focus only on a specific model family or application domain, this paper compares Transformer architectures across multiple dimensions, including architecture type, training objective, attention or sequence modeling mechanism, computational complexity, scalability, context length, modality, openness, interpretability, safety, deployment suitability, and benchmark behavior.

The major contributions of this paper are as follows:

1. It presents a structured taxonomy of Transformer architectures from the original encoder-decoder model to modern foundation models and emerging alternatives.
2. It compares encoder-only, decoder-only, encoder-decoder, efficient, vision, multimodal, Mixture-of-Experts, open-weight, and alternative sequence modelling architectures.
3. It discusses the strengths, limitations, and application suitability of major Transformer families.
4. It identifies key challenges related to computational cost, quadratic attention complexity, hallucination, interpretability, bias, safety, reproducibility, and evaluation limitations.
5. It highlights major research gaps and future directions for efficient, trustworthy, multimodal, long-context, and domain-specific Transformer-based systems.

It introduces a benchmark-based discussion that connects architectural choices with evaluation evidence from language understanding, reasoning, coding, long-context, multimodal, and holistic evaluation benchmarks.

It revises and verifies the reference list by prioritizing peer-reviewed papers, arXiv records with identifiers, official technical reports, model cards, and benchmark repositories.

1.1 Research Questions

This review is guided by the following research questions:

RQ1: How have Transformer architectures evolved from the original Transformer model to modern foundation models and emerging alternatives?

RQ2: What are the major architectural differences among encoder-only, decoder-only, encoder-decoder, efficient, vision, multimodal, Mixture-of-Experts, open-weight, and alternative Transformer-based models?

RQ3: What are the main strengths, limitations, and application areas of different Transformer architecture families?

RQ4: How do Transformer models differ in terms of attention mechanism, computational complexity, context length, scalability, modality support, openness, and deployment suitability?

RQ5: What are the major challenges and research gaps in Transformer architectures

related to efficiency, interpretability, hallucination, safety, reproducibility, and long-context reasoning?

RQ6: What future research directions can improve the design, evaluation, and real-world deployment of Transformer-based and hybrid artificial intelligence systems?

RQ7: What benchmark-based evidence supports the comparative strengths, limitations, and research gaps of major Transformer and hybrid architecture families?

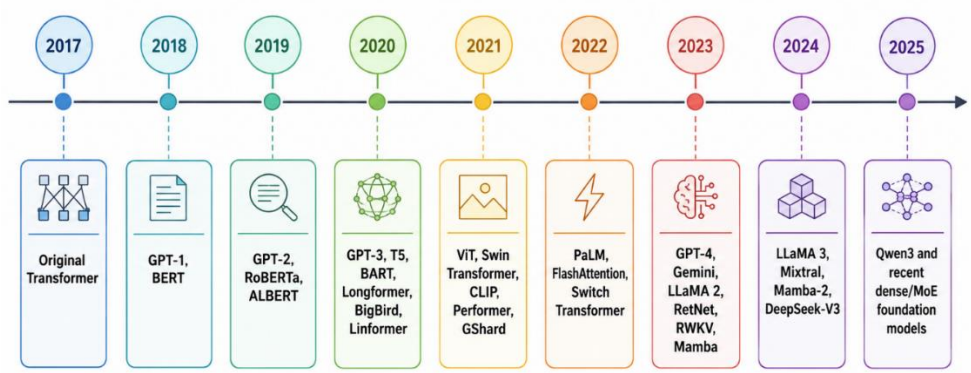

Figure 1. Timeline of Transformer Evolution from 2017 to 2025

2. Review Methodology

This study adopts a structured narrative literature review design. A fully statistical meta-analysis was not appropriate because Transformer papers differ substantially in task type, model scale, training data disclosure, benchmark protocol, modality, licensing, and availability of reproducible artifacts. However, to address methodological rigor, the review follows a transparent search, screening, coding, and synthesis process. The purpose is to compare architectural families rather than to pool a single performance metric.

2.1 Search Sources and Time Span

The literature search covered peer-reviewed venues, preprints, technical reports, and official model documentation published from 2017 to 2025. Sources included IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, Google Scholar, arXiv, OpenReview, model cards, official technical reports, and public benchmark repositories. The time span begins with the original Transformer and extends to recent dense, MoE, multimodal, long-context, reasoning, and hybrid architectures.

2.2 Search Keywords

The main search terms were: Transformer architecture, self-attention, BERT, GPT, encoder-only Transformer, decoder-only Transformer, encoder-decoder Transformer, efficient Transformer, sparse attention, FlashAttention, long-context Transformer, Vision Transformer, multimodal Transformer, Mixture-of-Experts, open-weight foundation model, GPT-4o, Gemini, Llama, DeepSeek, Qwen, Mamba, RetNet, RWKV, Jamba, Griffin, RecurrentGemma, Hyena, xLSTM, LongBench, RULER, HELM, MMLU, HumanEval, SWE-bench, and MMMU.

Table 1. Structured search, screening, and synthesis protocol

Stage	Procedure	Purpose
Identification	Search academic databases, arXiv/OpenReview, official reports, model cards, and benchmark repositories from 2017-2025.	Capture foundational and recent Transformer-related work.
Screening	Remove duplicates, non-technical opinion pieces, unsupported claims, and studies without architectural or evaluation relevance.	Improve source quality and relevance.
Eligibility	Retain papers that introduce a model family, attention mechanism, scaling method, benchmark, multimodal approach, or hybrid alternative.	Ensure each source contributes to architecture comparison.
Coding	Code each source by architecture type, objective, attention/sequence mechanism, complexity, modality, openness, benchmark evidence, and deployment constraints.	Support structured comparison instead of purely descriptive summary.
Synthesis	Compare evidence across model families, identify benchmark patterns, and map gaps to cited studies.	Link conclusions to literature and benchmark evidence.

2.3 Inclusion and Exclusion Criteria

Studies were included when they introduced a major architecture, proposed a significant attention or sequence modeling mechanism, improved computational efficiency, extended Transformers to a new modality, reported influential foundation-model capabilities, provided open-weight or reproducible artifacts, or introduced a widely used benchmark. Studies were excluded when they were duplicate reports, non-technical blog posts, short opinion articles, papers without architectural contribution, or works not directly related to Transformer-based, Transformer-inspired, or sequence modeling alternatives.

2.4 Data Extraction and Coding Framework

For each selected source, the review extracted the model family, architecture type, training objective, attention or sequence mechanism, computational complexity, supported context length, modality, openness, benchmark evidence, deployment cost, reproducibility, and known limitations. This coding framework directly supports the comparative analysis presented in Section 13.

2.5 Quality Assessment

Each source was assessed using four criteria: architectural novelty, technical clarity, evaluation evidence, and reproducibility. Papers with strong architectural details and benchmark evidence were prioritized. Closed-source model reports were included only when they represented influential frontier developments, but their limitations for reproducibility are explicitly discussed. This distinction is important because closed models may report strong benchmark results while withholding training data, model

weights, and full evaluation details.

2.6 Reference Verification and Source Reliability

To address concerns about reference originality and scholarly reliability, the revised manuscript classifies sources by reliability level. Peer-reviewed conference and journal papers were treated as the strongest evidence for established architectures and benchmarks. arXiv preprints were retained when they introduced influential models or benchmarks not yet available in final proceedings, but they were cited with explicit arXiv identifiers. Official model cards, system cards, and technical reports were used for frontier or closed models where peer-reviewed architectural details are unavailable. Non-technical blogs and secondary media sources were excluded from the manuscript bibliography unless they were official release pages from the model developer. This process improves traceability while acknowledging that many recent foundation-model releases are documented first through preprints, system cards, and official technical reports.

Table 2. Source verification and reliability assessment protocol

Source category	Use in review	Reliability treatment
Peer-reviewed papers	Foundational architectures, benchmarks, and accepted conference/journal work.	Preferred evidence source when available.
arXiv / OpenReview preprints	Recent architectures, benchmarks, and technical reports pending formal publication.	Included with arXiv identifier and used cautiously.
Official model/system cards	Closed or frontier models without full peer-reviewed disclosure.	Used for capabilities, safety, and release details; limitations noted.
Benchmark repositories	Dataset size, task categories, protocols, and leaderboards.	Used to verify benchmark scope and reproducibility artifacts.
Blogs / media / opinion pieces	Not used as scholarly evidence.	Excluded unless official primary release documentation.

2.7 Synthesis Strategy

The synthesis is organized by model family and then evaluated through benchmark-informed comparison. Instead of ranking models by a single leaderboard score, the review compares how different architectures perform across task categories: language understanding, generation, code, long-context reasoning, multimodal reasoning, efficiency, and safety. This approach avoids overgeneralization because a model that performs strongly on MMLU may still fail on long-context retrieval, multimodal diagram understanding, software engineering issue resolution, or safety-sensitive deployment.

2.8 Related Review Studies

Several review studies have examined Transformer architectures from different perspectives. Lin et al. provided a broad survey of Transformer variants and organized them according to architectural modifications, pre-training methods, and application areas [34]. Tay et al. focused on efficient Transformer models and reviewed approaches

for reducing the computational and memory complexity of self-attention [35]. Zhao et al. surveyed large language models, training strategies, resources, applications, and challenges [36]. Bommasani et al. introduced the broader concept of foundation models and discussed capabilities, risks, and societal implications [37].

The present study differs from these reviews in four ways. First, it updates the scope to include 2024-2025 developments such as GPT-4o, Gemini 2.5, Llama 4, DeepSeek-R1, Qwen3, Jamba, RecurrentGemma, Hyena, and xLSTM. Second, it adds benchmark-based evaluation rather than relying only on model descriptions. Third, it compares closed, open-weight, dense, MoE, multimodal, long-context, and hybrid architectures within a single framework. Fourth, it maps research gaps to cited benchmark and architectural evidence.

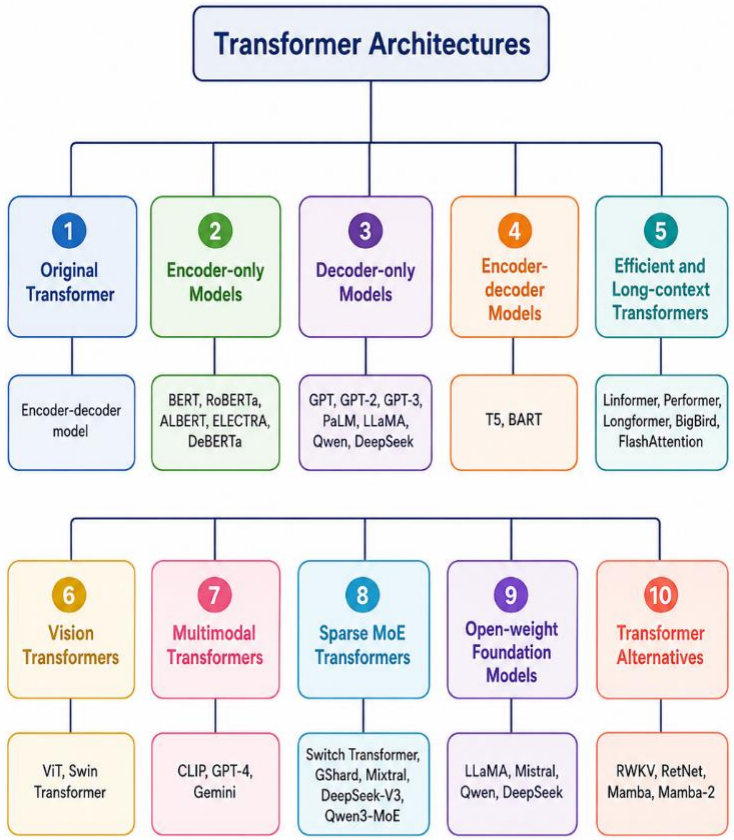

Figure 2. Taxonomy of Transformer Architectures

3. Original Transformer Architecture

The original Transformer was introduced in the paper "Attention Is All You Need" by Vaswani et al. [1]. The model was designed for sequence transduction tasks, particularly machine translation. It used an encoder-decoder structure in which the encoder processes the input sequence and the decoder generates the output sequence. The most important innovation of the model was multi-head self-attention, which allows the model to learn different types of relationships among tokens.

The Transformer consists of several major components: input embeddings, positional encodings, multi-head attention, feed-forward neural networks, residual connections, and layer normalization. Since the architecture does not use recurrence, positional encoding is required to represent token order. Multi-head attention enables the model to attend to information from different representation subspaces, improving its ability to capture syntactic and semantic relationships.

The main strength of the original Transformer is parallelization. Unlike recurrent neural networks, which process tokens sequentially, Transformers process all tokens simultaneously during training. This makes them highly suitable for large-scale training. However, the standard self-attention mechanism has quadratic time and memory complexity with respect to sequence length. As a result, processing very long sequences becomes computationally expensive.

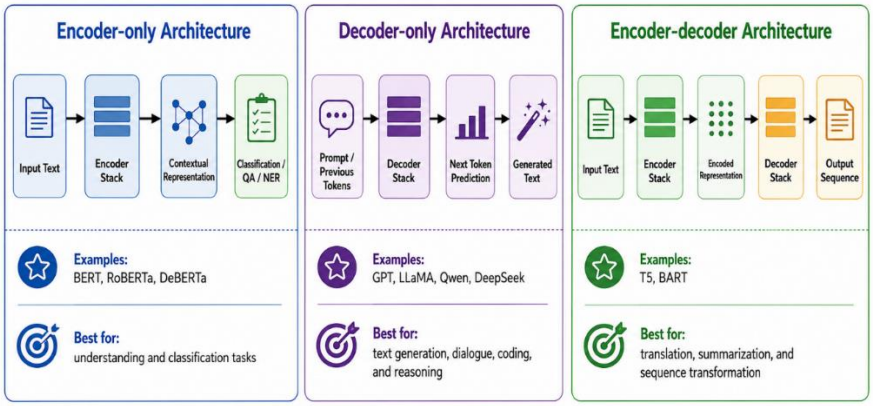

Figure 3. Encoder-only vs Decoder-only vs Encoder-decoder Transformer Architectures

4. Encoder-Only Transformer Models

Encoder-only Transformers are mainly used for language understanding and classification tasks. BERT is the most influential model in this category [2]. It introduced bidirectional pre-training using masked language modeling and next sentence prediction. Unlike autoregressive models that read text from left to right, BERT learns contextual representations by considering both left and right context. This made BERT highly effective for tasks such as sentiment analysis, question answering, named entity recognition, text classification, and natural language inference.

RoBERTa improved BERT by optimizing its pre-training strategy [3]. It removed the next sentence prediction task, used larger mini-batches, applied dynamic masking, and trained on more data. ALBERT reduced model size using parameter sharing and factorized embeddings [4]. ELECTRA introduced replaced-token detection, where the model learns to distinguish original tokens from replaced tokens [5]. DeBERTa improved BERT by separating content and positional information through disentangled attention [6].

The strength of encoder-only models is their ability to generate rich contextual representations. They are particularly effective for discriminative tasks where the goal is to classify or understand input text. However, they are not naturally designed for open-ended text generation. Therefore, while BERT-style models remain useful for

classification and understanding, they are less suitable for dialogue, story generation, code generation, and long-form reasoning.

5. Decoder-Only Generative Transformer Models

Decoder-only Transformers are the foundation of modern generative AI. These models are trained using autoregressive next-token prediction, where the model predicts the next token based on previous tokens. GPT-1 demonstrated the effectiveness of generative pre-training followed by fine-tuning [7]. GPT-2 showed that large-scale language models could perform multiple tasks in a zero-shot setting [8].

GPT-3 marked a major milestone in large language model research. With 175 billion parameters, GPT-3 demonstrated few-shot and in-context learning, where the model can perform new tasks using only prompts or examples without parameter updates [9]. This changed the direction of NLP research from task-specific fine-tuning toward prompt-based learning and general-purpose language modeling.

PaLM further demonstrated the benefits of scaling language models for reasoning, multilingual processing, and code-related tasks [10]. GPT-4 extended the decoder-only Transformer family into multimodal learning by accepting both text and image inputs and generating text outputs [11]. Modern decoder-only models such as LLaMA, Mistral, Qwen, and DeepSeek have expanded this research direction by offering open-weight models, improved reasoning, longer context windows, and efficient inference designs [25][29].

The major strength of decoder-only Transformers is their generative capability. They are widely used for text generation, summarization, dialogue systems, code generation, tutoring systems, reasoning, and agentic workflows. However, they also have limitations, including hallucination, high inference cost, prompt sensitivity, bias, safety risks, and limited interpretability.

6. Encoder-Decoder Transformer Models

Encoder-decoder Transformers are suitable for sequence-to-sequence tasks where the input and output are both sequences. T5 introduced a unified text-to-text framework in which all NLP tasks are converted into text input and text output [12]. This approach enabled a single model design to handle translation, summarization, classification, and question answering.

BART combined bidirectional encoding with autoregressive decoding using denoising pre-training [13]. It corrupted input text and trained the model to reconstruct the original version. This made BART effective for text generation and comprehension tasks.

Compared with encoder-only models, encoder-decoder models are more flexible for transformation tasks. Compared with decoder-only models, they explicitly encode the input before generating the output. However, they are computationally more complex because both encoder and decoder components are used. In modern applications, decoder-only models dominate general-purpose generative AI, while encoder-decoder models remain useful for controlled text transformation, translation, and summarization.

7. Efficient and Long-Context Transformers

A major limitation of the standard Transformer is the quadratic complexity of self-attention. If the input sequence length increases, the computational and memory cost

grows rapidly. This limitation is serious for long documents, legal records, scientific articles, software repositories, and retrieval-augmented systems.

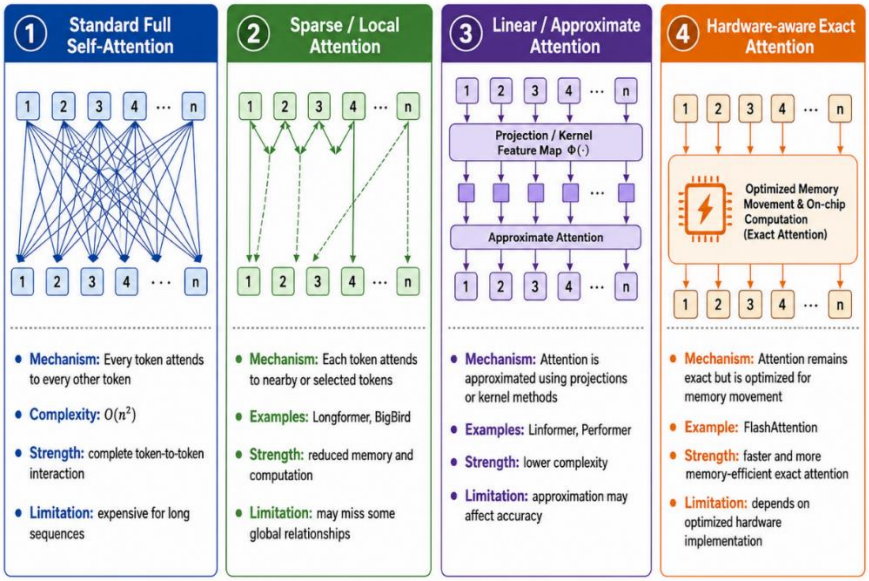

Figure 4. Standard Attention vs Efficient/Sparse Attention

Several models were proposed to solve this problem. Linformer reduced attention complexity by using low-rank projections [14]. Performer used random feature approximations to estimate softmax attention in linear time [15]. Longformer introduced a combination of sliding-window local attention and global attention, making it suitable for long-document classification and question answering [16]. BigBird used sparse attention patterns and showed that sparse attention can preserve important theoretical properties of full attention [17].

FlashAttention took a different approach. Instead of approximating attention, it optimized exact attention computation by reducing memory movement between GPU memory levels [18]. This hardware-aware design improved speed and memory efficiency and became important for training and inference of modern long-context models.

Efficient Transformers are useful for long-context applications. However, sparse or approximate attention may introduce trade-offs in accuracy, generalization, and implementation complexity. Therefore, efficient attention remains an active research area.

8. Vision Transformers

Transformers were first designed for NLP but later became important in computer vision. Vision Transformer divided images into fixed-size patches and treated them as token sequences [19]. This allowed the standard Transformer architecture to be applied directly to image classification. ViT showed that with sufficient training data, Transformers can compete with or outperform convolutional neural networks.

Swin Transformer improved the vision Transformer design by introducing hierarchical feature representation and shifted window attention [20]. This made it more suitable for dense prediction tasks such as object detection and semantic segmentation.

The main advantage of vision Transformers is their ability to model global relationships in images. Unlike convolutional networks, which rely heavily on local receptive fields, Transformers can capture long-range visual dependencies. However, they often require large datasets, strong regularization, and significant computational resources.

9. Multimodal Transformers

Multimodal Transformers combine information from different data types such as text, images, audio, video, and code. CLIP aligned image and text representations using contrastive learning [21]. This enabled zero-shot image classification using natural language prompts and demonstrated the power of connecting visual and textual representations.

GPT-4 and Gemini represent more advanced multimodal Transformer-based systems [11], [22]. These models can process multiple input types and perform complex reasoning across modalities. Multimodal Transformers are important for applications such as visual question answering, image captioning, video understanding, medical imaging, robotics, educational tutoring, and human-computer interaction.

Despite their strengths, multimodal Transformers face challenges. They require large and diverse datasets, careful alignment between modalities, high computational resources, and reliable evaluation methods. Moreover, many frontiers multimodal models are closed-source, making scientific comparison and reproducibility difficult.

10. Sparse Mixture-of-Experts Transformers

Sparse Mixture-of-Experts models are designed to increase model capacity without activating all parameters for every input. Instead of using the same feed-forward network for all tokens, MoE models route tokens to selected expert networks. This allows models to have a very large total number of parameters while keeping active computation relatively low.

Switch Transformer simplified MoE routing and showed that sparse expert models could scale to very large parameter counts [23]. GShard also demonstrated the use of conditional computation and automatic sharding for large multilingual models [24]. Mixtral applied sparse MoE design to open-weight decoder-only models [25]. DeepSeek-V3 used a large MoE architecture with 671 billion total parameters and 37 billion activated parameters per token [26]. Qwen3 also introduced both dense and MoE models with multilingual and reasoning-oriented capabilities [27].

MoE Transformers are attractive because they improve the balance between capacity and computational cost. However, they introduce challenges such as routing instability, load balancing, expert specialization, distributed training complexity, and deployment difficulty.

11. Open-Weight Foundation Models

Open-weight foundation models have become important for academic research and real-world deployment. LLaMA 2 released open foundation and chat models in different sizes [28]. LLaMA 3 expanded this direction with models supporting multilinguality, coding, reasoning, tool use, and long-context processing [29]. Mistral and Mixtral contributed efficient open-weight architectures using sliding-window attention and sparse MoE designs [25].

Open-weight models are valuable because they support transparency, reproducibility,

fine-tuning, and domain adaptation. Researchers can evaluate model behavior, apply models to specialized domains, and develop safer deployment pipelines. However, openness also creates risks related to misuse, privacy, copyright, and harmful content generation.

12. Recent Transformer Alternatives and Hybrid Architectures

Although Transformers dominate modern AI, recent research has increasingly explored alternatives and hybrid architectures to reduce the cost of full self-attention and improve long-context efficiency. RWKV combines Transformer-style parallel training with recurrent inference [30]. RetNet introduced a retention mechanism that supports parallel training and efficient inference [31]. Mamba introduced selective state-space models with linear scaling in sequence length [32], while Mamba-2 connected state-space models and Transformers through structured state-space duality [33].

Recent hybrid systems extend this trend. Jamba combines Transformer layers, Mamba layers, and sparse MoE routing to provide high throughput and reduced memory usage while retaining strong benchmark performance on standard and long-context evaluations [42]. RecurrentGemma is based on the Griffin architecture, which combines gated linear recurrences with local sliding-window attention for efficient long-sequence processing [43]. Hyena replaces attention with implicitly parameterized long convolutions and data-controlled gating, targeting subquadratic long-sequence modeling [45]. xLSTM revisits recurrent neural networks by adding exponential gating, scalar and matrix memory variants, and residual xLSTM blocks, showing that scaled recurrent architectures remain competitive in modern sequence modeling [44].

These alternatives do not eliminate the importance of Transformers; rather, they show that the next generation of sequence models is likely to be hybrid. Full attention remains strong for flexible token-to-token interaction, but state-space, recurrent, retention, convolutional, retrieval, and sparse expert components can improve throughput, memory use, streaming ability, and long-context deployment. The key research question is no longer whether attention should be replaced entirely, but how attention should be combined with other mechanisms for different deployment constraints.

Table 3. Additional recent architectures incorporated in the revised review

Recent development	Architecture family	Main contribution	Why it matters for this review
GPT-4o	Multimodal decoder-style foundation model	Real-time reasoning across text, vision, and audio modalities.	Strengthens discussion of native multimodal and interactive AI.
Gemini 2.5	Multimodal thinking model	Emphasizes advanced reasoning, multimodal understanding, and long-context input.	Shows movement toward long-context multimodal reasoning.
Llama 4	Open-weight multimodal MoE	Introduces native multimodality and MoE design in the Llama family.	Important for open-weight multimodal comparison.
DeepSeek-R1	Reasoning-oriented LLM	Uses reinforcement learning and cold-start	Represents post-training and reasoning-focused model

		data for mathematical, code, and reasoning tasks.	development.
Qwen3	Dense and MoE LLM family	Provides multilingual, reasoning, and efficiency-oriented dense/MoE variants.	Adds 2025 open-weight dense/MoE evidence.
Jamba/Jamba -1.5	Transformer-Mamba-MoE hybrid	Interleaves attention, Mamba, and MoE layers for long-context efficiency.	Illustrates hybrid design beyond pure attention.
RecurrentGe mma/Griffin	Recurrent-local attention hybrid	Combines gated linear recurrence with local attention.	Shows efficient alternatives for long sequences.
Hyena and xLSTM	Convolutional/re current alternatives	Use long convolutions or scaled LSTM memory mechanisms.	Broadens the review beyond attention-only systems.

13. Comparative and Benchmark-Based Analysis of Transformer Families

The core contribution of this review is the comparison of Transformer families across architecture, efficiency, benchmark behavior, and deployment suitability. A high-level table alone is insufficient because benchmark strength depends on task type. For example, an encoder-only model can be strong on classification benchmarks but unsuitable for open-ended generation, while a decoder-only model can generate fluent text but may hallucinate or fail on long-context factual retrieval. Therefore, this section expands the comparison through architectural, benchmark, and deployment dimensions.

Table 4. Expanded comparative analysis of Transformer and hybrid model families

Model family	**Mechanism and objective**	**Efficiency/context profile**	**Benchmark pattern**	**Deployment implication**
Original Transformer	Encoder-decoder self-attention for sequence transduction.	Quadratic attention; strong parallel training but costly for long sequences.	Strong foundation for translation and sequence modeling.	Useful as conceptual baseline; rarely deployed unchanged at frontier scale.
Encoder-only	Bidirectional masked language modeling; representation learning.	Moderate inference cost; fixed-length input windows.	Strong on classification, QA, NER, GLUE/SuperGLUE-style tasks.	Best for understanding, retrieval ranking, classification, and domain classifiers.
Decoder-only	Autoregressive next-token prediction; instruction tuning and RL	High inference cost; KV-cache grows with context length.	Strong on MMLU, coding, reasoning, dialogue, and generation benchmarks.	Best for assistants, code generation, agents,

	alignment.			tutoring, and general generation.
Encoder-decoder	Input encoder plus autoregressive decoder for sequence transformation.	More components than encoder-only; controlled output generation.	Strong on summarization, translation, and transformation benchmarks.	Useful when source sequence must be explicitly encoded before output.
Efficient/long-context	Sparse, local, approximate, or hardware-aware exact attention.	Reduces memory/time or improves practical attention throughput.	Evaluated on LongBench, RULER, document QA, and long summarization.	Useful for legal, scientific, codebase, and RAG systems.
Vision Transformer	Images as patches/tokens; global visual attention.	Requires large data and compute; hierarchical variants improve dense tasks.	Strong on ImageNet and downstream vision tasks when trained at scale.	Useful for image classification, detection, segmentation, and medical images.
Multimodal Transformer	Aligns or jointly processes text, image, audio, video, and code.	High data and alignment cost; evaluation more difficult.	Evaluated on VQA, MMMU, image-text retrieval, and multimodal reasoning.	Useful for tutoring, robotics, visual QA, document understanding, and HCI.
Sparse MoE	Routes tokens to selected expert networks.	Large total parameters with lower active computation.	Strong scale-quality trade-off but routing and serving are complex.	Useful for frontier models when distributed infrastructure is available.
Open-weight foundation models	Dense or MoE models with publicly released weights.	Efficiency varies by size and quantization.	Enable reproducible benchmarking and domain adaptation.	Useful for academic research, local deployment, fine-tuning, and privacy-sensitive use.
Hybrid alternatives	Attention combined with SSM, recurrence, retention, convolution, or MoE.	Targets lower memory use and better long-context throughput.	Evaluated on language modeling, long-context, and standard LLM benchmarks.	Promising for resource-constrained and long-stream applications.

13.2 Benchmark-Based Discussion

Benchmark evidence must be interpreted by capability category rather than by a single overall score. MMLU evaluates broad academic and professional knowledge across 57 tasks [46], while MMLU-Pro was introduced because original MMLU scores began to saturate; it increases difficulty through reasoning-focused questions and ten answer options and reports performance drops of 16% to 33% compared with MMLU [51]. HELM argues for holistic evaluation across scenarios and metrics such as accuracy, calibration, robustness, fairness, bias, toxicity, and efficiency rather than a single benchmark number [47].

Long-context evaluation illustrates why nominal context length is not enough. LongBench covers long-document QA, multi-document QA, summarization, few-shot learning, synthetic tasks, and code completion across English and Chinese [48]. RULER further shows that simple needle-in-a-haystack retrieval can overestimate long-context ability because models can perform well on easy retrieval but degrade when task complexity and length increase [49]. Therefore, models claiming very large context windows should be evaluated for reasoning, aggregation, positional robustness, and lost-in-the-middle behavior rather than only maximum input length.

Coding benchmarks also show that architectural progress must be assessed through realistic tasks. HumanEval evaluates functional correctness of generated Python code using unit tests [52], while SWE-bench evaluates whether models can resolve real GitHub issues by editing real repositories [54]. SWE-bench demonstrates that solving practical software engineering issues requires long-context understanding, tool use, multi-file reasoning, and execution feedback, making it more demanding than isolated code generation.

Multimodal evaluation remains challenging. MMMU was designed to evaluate college-level multimodal understanding and reasoning across 11.5K questions covering six disciplines, 30 subjects, 183 subfields, and diverse image types [53]. MMMU-Pro further strengthens this by filtering questions answerable by text-only models and introducing vision-only settings [55]. These benchmarks reveal that multimodal models need not only image-text alignment, but also domain knowledge, diagram interpretation, OCR robustness, spatial reasoning, and cross-modal consistency.

Table 5. Benchmark categories used to strengthen the comparative analysis

Capability	Representative benchmarks	Most relevant model families	What the benchmark reveals	Main limitation
Language understanding	GLUE, SuperGLUE, SQuAD, MMLU, MMLU-Pro	Encoder-only and decoder-only models	Knowledge, classification, QA, reasoning breadth.	Can saturate; may not reflect real-world reliability.
Generation and reasoning	MMLU, GSM8K, MATH, BIG-Bench, HELM	Decoder-only and reasoning-tuned models	Problem solving, instruction following, reasoning	Prompt sensitivity and contamination risk.

			patterns.	
Code generation and software engineering	HumanEval, CodeXGLUE, MBPP, SWE-bench	Decoder-only code models and agentic LLMs	Functional correctness and real repository issue resolution.	HumanEval is narrow; SWE-bench requires tools and execution.
Long-context understanding	LongBench, RULER, Needle-in-Haystack	Efficient attention, long-context LLMs, hybrid SSM/recurrent models	Retrieval, aggregation, summarization, multi-hop reasoning, positional robustness.	Synthetic tasks may not capture all real uses.
Vision and multimodal reasoning	ImageNet, COCO, VQA, MMMU, MMMU-Pro	Vision Transformers and multimodal LLMs	Visual recognition, image-text alignment, diagram and expert reasoning.	Evaluation protocols differ and closed models limit reproducibility.
Safety and deployment reliability	HELM, toxicity/bias/robustness benchmarks, model cards	All foundation model families	Robustness, fairness, toxicity, calibration, efficiency, and safety.	Metrics are multidimensional and sometimes hard to aggregate.

Table 6. Quantitative benchmark evidence used for critical comparison

Benchmark	**Quantitative scope / reported evidence**	**Evaluation lesson for this review**
HELM [47]	30 language models, 42 scenarios, 7 metrics: accuracy, calibration, robustness, fairness, bias, toxicity, and efficiency.	Model comparison should be multi-metric; accuracy alone hides safety and efficiency trade-offs.
MMLU-Pro [51]	More than 12,000 questions across 14 domains; 10 answer choices; reported 16%-33% accuracy drop compared with MMLU.	Saturated benchmarks can overstate progress; harder reasoning-focused tests separate models more clearly.
LongBench [48]	21 datasets across 6 task categories in English and Chinese; includes QA, summarization, few-shot, synthetic, and code tasks.	Long context must be tested through diverse realistic tasks, not only nominal context length.
RULER [49]	17 long-context LMs, 13 tasks, configurable lengths; only about half maintained satisfactory performance at 32K tokens.	Needle retrieval is insufficient; long-context performance drops with length and task complexity.
SWE-bench [54]	2,294 real GitHub issues from 12 Python repositories; original	Coding ability requires repository-level reasoning,

	evaluation reported Claude 2 solved only 1.96%.	editing, testing, and environment interaction.
MMMU [53]	11.5K multimodal questions, 6 disciplines, 30 subjects, 183 subfields; GPT-4V and Gemini Ultra reached 56% and 59%.	Multimodal benchmarks reveal major gaps in expert-level visual-domain reasoning.

The quantitative evidence in Table 6 supports the reviewer-requested shift from description to evaluation. These benchmarks do not produce a single universal ranking; instead, they expose different capability dimensions. For instance, MMLU-Pro demonstrates that older knowledge benchmarks can become saturated, RULER shows that advertised context length can exceed effective context reasoning, SWE-bench demonstrates the difficulty of real software engineering tasks, and MMMU reveals that multimodal systems still struggle with expert-level visual reasoning.

13.3 Critical Evaluation of Model Families

Encoder-only models remain efficient and reliable for discriminative tasks, but their value decreases when tasks require long-form generation, dialogue, or tool use. Decoder-only models dominate modern generative AI because autoregressive pre-training scales well and supports prompting, instruction following, and agentic workflows. However, their strengths on reasoning and coding benchmarks do not remove hallucination, bias, prompt sensitivity, or high inference cost.

Efficient attention and long-context models address a genuine computational problem, but they should not be judged only by theoretical complexity. Sparse and approximate attention can reduce cost but may lose global interactions. Hardware-aware exact attention, such as FlashAttention, improves practical throughput but depends on implementation and hardware. Long-context benchmarks show that increasing context length does not automatically produce robust long-context reasoning.

Vision and multimodal Transformers broaden the scope of Transformers beyond text, but they introduce alignment, evaluation, and data-quality challenges. Multimodal models may perform well on image-text benchmarks while still failing on spatial reasoning, charts, diagrams, domain-specific images, or OCR-heavy tasks. MoE models provide a strong scale-efficiency trade-off, but routing instability, load balancing, expert specialization, and distributed serving complexity make deployment more difficult than dense models.

Open-weight models improve transparency, reproducibility, and domain adaptation, but openness also raises misuse, privacy, copyright, and safety concerns. Closed frontier models often report strong benchmark performance, but incomplete disclosure of training data, model weights, and evaluation protocols limits scientific verification. Hybrid models such as Jamba, RecurrentGemma, Hyena, and xLSTM are promising, yet their long-term role depends on whether they can match Transformer quality across broad tasks while providing clear efficiency advantages.

13.4 Architectural Trade-off Evaluation

A critical review should not treat architectural families as a simple timeline of improvement. Each family optimizes a different trade-off surface. Encoder-only models prioritize representation quality and stable classification, while decoder-only models prioritize general generation and instruction following at higher inference cost. Sparse

and approximate attention methods reduce complexity but may weaken global token interactions. Hardware-aware exact attention improves throughput but depends on GPU memory hierarchy. MoE increases total capacity while limiting active parameters, but it introduces routing, load-balancing, and distributed serving challenges. Hybrid state-space and recurrent models improve memory scaling and streaming behavior but may not yet match full-attention models on all open-ended reasoning tasks.

Table 8. Architectural trade-off matrix for model-family selection

Architecture choice	**Primary advantage**	**Main trade-off**	**Best suited deployment**
Encoder-only	Stable contextual representations and lower generation risk.	Limited open-ended generation and dialogue ability.	Classification, retrieval, NER, sentiment, domain detection.
Decoder-only dense	Strong general generation, reasoning, coding, and instruction following.	High inference cost, hallucination risk, KV-cache memory growth.	Assistants, tutoring, coding, agents, summarization.
Sparse / efficient attention	Lower long-sequence cost and better scalability.	Possible loss of global interactions or implementation complexity.	Documents, legal texts, scientific papers, RAG.
Hardware-aware exact attention	Maintains exact attention while improving speed and memory movement.	Benefits depend on specific accelerators and kernels.	Training/inference stacks with modern GPUs.
MoE	Higher total capacity with lower active compute.	Routing instability, expert imbalance, distributed deployment cost.	Large-scale multilingual and frontier foundation models.
State-space / recurrent hybrids	Linear or near-linear sequence scaling and streaming-friendly inference.	Still maturing ecosystem and mixed benchmark dominance.	Long streams, edge inference, low-memory long-context systems.
Multimodal models	Cross-modal reasoning over text, images, audio, video, and code.	Alignment difficulty, data quality, evaluation complexity, safety risks.	Education, robotics, medical imaging, visual QA, human-computer interaction.

This trade-off matrix strengthens the critical evaluation by connecting architecture design to practical model selection. The most suitable architecture depends on the task objective, context length, latency budget, available hardware, openness requirements, safety profile, and reproducibility expectations.

14. Key Challenges

Despite their success, Transformer and hybrid foundation models face several technical and practical challenges.

14.1 Computational Cost

Large Transformer models require massive computational resources for training and inference. This creates barriers for small institutions and researchers and raises environmental concerns due to energy consumption. Cost-aware comparison should include latency, memory, throughput, energy use, and hardware requirements, not only accuracy.

14.2 Quadratic Attention Complexity

Standard self-attention has quadratic complexity with respect to sequence length, making it expensive for long sequences. Efficient attention, sparse attention, FlashAttention, retrieval, state-space models, and recurrent hybrids reduce aspects of this cost, but long-context reasoning remains difficult.

14.3 Hallucination and Factuality

Generative Transformers can produce fluent but incorrect information. This is especially problematic in education, healthcare, law, finance, and scientific research. Benchmark scores must therefore be supplemented with factuality, citation, retrieval, and human evaluation protocols.

14.4 Interpretability

Transformers are difficult to interpret. Attention weights do not always provide faithful explanations of model decisions. Reliable interpretability requires feature attribution, counterfactual analysis, probing, mechanistic interpretability, and task-specific explanation methods.

14.5 Bias and Safety

Transformer models learn from large-scale datasets that may contain social, cultural, gender, religious, political, and linguistic biases. These biases can appear in outputs and may be amplified in downstream applications. Safety evaluation must include toxicity, fairness, robustness, privacy, refusal behavior, and misuse potential.

14.6 Reproducibility and Openness

Many frontier models are closed-source and provide limited information about training data, architecture, alignment procedure, and evaluation. This makes fair comparison difficult. Open-weight models improve reproducibility but may differ in safety mechanisms and documentation quality.

14.7 Evaluation Limitations

Benchmark performance does not always reflect real-world reliability. Benchmarks can saturate, leak into training data, vary by prompt format, and fail to capture deployment context. Model evaluation should therefore combine standardized benchmarks, stress tests, domain-specific datasets, human evaluation, and operational metrics.

15. Evidence-Backed Research Gaps

The revised review maps research gaps to benchmark and architectural evidence rather

than presenting them only as author observations.

Table 7. Evidence-backed research gaps derived from architecture and benchmark analysis

Research gap	Evidence from reviewed literature/ benchmarks	Affected model families	Implication
Cost-aware comparison is weak	Many reports emphasize benchmark accuracy while underreporting latency, memory, energy, and hardware requirements.	Large decoder-only, multimodal, MoE, and closed models	Future studies should report accuracy-efficiency trade-offs.
Long-context reasoning is not solved	LongBench and RULER show that longer context windows do not guarantee robust retrieval, aggregation, or multi-hop reasoning.	Long-context LLMs, efficient attention, SSM/recurrent hybrids	Evaluate usable context length, not only maximum context length.
Benchmark saturation reduces discriminative value	MMLU-Pro was proposed because MMLU became less discriminative for strong models and required harder reasoning-focused questions.	Frontier decoder-only and reasoning models	Evaluation must evolve as models improve.
Real software engineering remains difficult	SWE-bench requires multi-file edits and issue resolution, exposing limitations beyond HumanEval-style code generation.	Code LLMs and agentic systems	Model evaluation should include tools, execution feedback, and repository-scale reasoning.
Multimodal reasoning needs stronger evaluation	MMMU and MMMU-Pro show that expert visual reasoning requires more than image-text alignment or OCR.	Vision and multimodal Transformers	Benchmarks should include diagrams, charts, domain images, and vision-only settings.
Closed vs open comparison remains unfair	Closed models often have incomplete disclosure, while open-weight models enable reproducibility but vary in quality and safety.	Closed frontier models and open-weight models	Comparisons should separate capability, openness, cost, and reproducibility.
Interpretability remains limited	Attention visualization alone is not a faithful explanation method for model decisions.	All Transformer families	Need reliable explanation, probing, and mechanistic analysis.

Hybrid architectures need broader validation	Jamba, RecurrentGemma, Hyena, and xLSTM show efficiency promise but require broader task and deployment evaluation.	Hybrid SSM/recurrent/convolutional architectures	Future work should test whether efficiency gains generalize across domains.

16. Future Research Directions

Future Transformer research should move from single-score benchmark competition toward multi-dimensional evaluation. A model should be evaluated not only by accuracy, but also by latency, memory, cost, reproducibility, safety, calibration, robustness, and suitability for the target domain.

Efficient long-context modeling remains a major direction. Future work should combine attention, retrieval, memory, state-space models, recurrence, compression, and hardware-aware kernels to process long documents, code repositories, videos, and scientific literature without excessive cost.

Hybrid architecture design will become increasingly important. Future models may combine Transformers with Mamba-style state-space layers, recurrent memory, retention mechanisms, local attention, sparse expert routing, and retrieval modules. Such models should be tested across both standard and long-context benchmarks.

Multimodal reasoning should move beyond image-text matching toward diagram understanding, chart reasoning, video reasoning, spatial reasoning, audio-visual alignment, and robotics. Benchmarks should evaluate cross-modal consistency and domain-specific multimodal reasoning.

Trustworthy AI remains critical. Transformer-based systems need better factuality, safety alignment, bias reduction, interpretability, robustness, and privacy protection. These requirements are especially important for education, healthcare, law, finance, software engineering, and public-sector applications.

Open and reproducible research should be strengthened through model cards, dataset documentation, training details, benchmark protocols, inference cost reporting, and open evaluation harnesses. Domain-specific Transformers and lightweight models will also become more important for edge devices, mobile applications, classroom tools, medical systems, and software engineering platforms.

17. Conclusion

This paper presented a revised comparative literature review of Transformer architectures from the original Transformer to modern foundation models and emerging hybrid alternatives. In response to reviewer comments, the methodology was strengthened through explicit search sources, inclusion and exclusion criteria, coding dimensions, source reliability checks, quality assessment, and synthesis procedures. The comparative analysis was expanded beyond a brief table and now includes architecture-level, benchmark-based, quantitative, critical, and deployment-oriented comparison.

The review shows that encoder-only models remain strong for understanding and classification, decoder-only models dominate generative AI and reasoning, encoder-decoder models remain useful for sequence transformation, efficient Transformers

address long-context limitations, vision Transformers extend token-based modeling to images, multimodal Transformers enable cross-modal reasoning, MoE models improve scale-efficiency trade-offs, and open-weight models support reproducibility and domain adaptation. Recent architectures such as GPT-4o, Gemini 2.5, Llama 4, DeepSeek-R1, Qwen3, Jamba, RecurrentGemma, Hyena, and xLSTM demonstrate that the field is moving toward multimodal, reasoning-focused, efficient, and hybrid AI systems.

Benchmark-based analysis indicates that no single model family is universally best. MMLU/MMLU-Pro, HELM, HumanEval, SWE-bench, LongBench, RULER, and MMMU each reveal different strengths and limitations, while the revised reference list improves traceability by prioritizing peer-reviewed papers, arXiv identifiers, official model/system cards, and benchmark repositories. Therefore, future evaluation should be capability-specific, cost-aware, reproducible, and domain-sensitive. Overall, Transformers remain the dominant architecture in modern AI, but the next generation of models is likely to combine attention with state-space, recurrence, retrieval, memory, and sparse expert mechanisms to improve scalability, reliability, and practical usability.

Declarations
Competing
Interests

The authors declare that they have no competing interests.

Authors' Contribution

All the authors have contributed in the paper.

References

[1] A. Vaswani et al., "Attention is all you need," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional Transformers for language understanding," in Proc. NAACL-HLT, 2019.

[3] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," arXiv:1907.11692, 2019.

[4] Z. Lan et al., "ALBERT: A lite BERT for self-supervised learning of language representations," in Proc. International Conference on Learning Representations (ICLR), 2020.

[5] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," in Proc. International Conference on Learning Representations (ICLR), 2020.

[6] P. He, X. Liu, J. Gao, and W. Chen, "DeBERTa: Decoding-enhanced BERT with disentangled attention," in Proc. International Conference on Learning Representations (ICLR), 2021.

[7] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI Technical Report, 2018.

[8] A. Radford et al., "Language models are unsupervised multitask learners," OpenAI Technical Report, 2019.

[9] T. B. Brown et al., "Language models are few-shot learners," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2020.

[10] A. Chowdhery et al., "PaLM: Scaling language modeling with Pathways," Journal of Machine Learning Research, vol. 24, no. 240, pp. 1-113, 2023; arXiv:2204.02311.

[11] OpenAI, "GPT-4 technical report," arXiv:2303.08774, 2023.

[12] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text Transformer," Journal of Machine Learning Research, vol. 21, no. 140, pp. 1-67, 2020.

[13] M. Lewis et al., "BART: Denoising sequence-to-sequence pre-training for natural language

generation, translation, and comprehension," in Proc. ACL, 2020.
[14] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," arXiv:2006.04768, 2020.
[15] K. Choromanski et al., "Rethinking attention with Performers," in Proc. International Conference on Learning Representations (ICLR), 2021.
[16] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document Transformer," arXiv:2004.05150, 2020.
[17] M. Zaheer et al., "Big Bird: Transformers for longer sequences," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2020.
[18] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Re, "FlashAttention: Fast and memory-efficient exact attention with IO-awareness," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2022.
[19] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in Proc. International Conference on Learning Representations (ICLR), 2021.
[20] Z. Liu et al., "Swin Transformer: Hierarchical vision Transformer using shifted windows," in Proc. IEEE/CVF International Conference on Computer Vision (ICCV), 2021.
[21] A. Radford et al., "Learning transferable visual models from natural language supervision," in Proc. International Conference on Machine Learning (ICML), 2021.
[22] Gemini Team, "Gemini: A family of highly capable multimodal models," arXiv:2312.11805, 2023.
[23] W. Fedus, B. Zoph, and N. Shazeer, "Switch Transformers: Scaling to trillion parameter models with simple and efficient sparsity," Journal of Machine Learning Research, vol. 23, no. 120, pp. 1-39, 2022.
[24] D. Lepikhin et al., "GShard: Scaling giant models with conditional computation and automatic sharding," in Proc. International Conference on Learning Representations (ICLR), 2021.
[25] A. Q. Jiang et al., "Mixtral of experts," arXiv:2401.04088, 2024.
[26] DeepSeek-AI, "DeepSeek-V3 technical report," arXiv:2412.19437, 2024.
[27] A. Yang et al., "Qwen3 technical report," arXiv:2505.09388, 2025.
[28] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," arXiv:2307.09288, 2023.
[29] A. Grattafiori et al., "The Llama 3 herd of models," arXiv:2407.21783, 2024.
[30] B. Peng et al., "RWKV: Reinventing RNNs for the Transformer era," arXiv:2305.13048, 2023.
[31] Y. Sun et al., "Retentive network: A successor to Transformer for large language models," arXiv:2307.08621, 2023.
[32] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," arXiv:2312.00752, 2023.
[33] T. Dao and A. Gu, "Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality," arXiv:2405.21060, 2024.
[34] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of Transformers," AI Open, vol. 3, pp. 111-132, 2022.
[35] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient Transformers: A survey," ACM Computing Surveys, vol. 55, no. 6, pp. 1-28, 2022.
[36] W. X. Zhao et al., "A survey of large language models," arXiv:2303.18223, 2023.
[37] R. Bommasani et al., "On the opportunities and risks of foundation models," arXiv:2108.07258, 2021.
[38] OpenAI, "GPT-4o system card," OpenAI, 2024. Available: https://openai.com/index/gpt-4o-system-card/.
[39] D. Guo et al., "DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning," arXiv:2501.12948, 2025.
[40] G. Comanici et al., "Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities," arXiv:2507.06261, 2025.
[41] Meta AI, "The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation," Meta AI Blog, 2025. Available: https://ai.meta.com/blog/llama-4-multimodal-intelligence/.

[42] O. Lieber et al., "Jamba: A hybrid Transformer-Mamba language model," arXiv:2403.19887, 2024.
[43] A. Botev et al., "RecurrentGemma: Moving past Transformers for efficient open language models," arXiv:2404.07839, 2024.
[44] M. Beck et al., "xLSTM: Extended Long Short-Term Memory," arXiv:2405.04517, 2024.
[45] M. Poli et al., "Hyena hierarchy: Towards larger convolutional language models," arXiv:2302.10866, 2023.
[46] D. Hendrycks et al., "Measuring massive multitask language understanding," in Proc. International Conference on Learning Representations (ICLR), 2021.
[47] P. Liang et al., "Holistic evaluation of language models," arXiv:2211.09110, 2022.
[48] Y. Bai et al., "LongBench: A bilingual, multitask benchmark for long context understanding," in Proc. ACL, 2024; arXiv:2308.14508.
[49] C.-P. Hsieh et al., "RULER: What is the real context size of your long-context language models?" arXiv:2404.06654, 2024.
[50] Y. Bai et al., "LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks," arXiv:2412.15204, 2024.
[51] Y. Wang et al., "MMLU-Pro: A more robust and challenging multi-task language understanding benchmark," arXiv:2406.01574, 2024.
[52] M. Chen et al., "Evaluating large language models trained on code," arXiv:2107.03374, 2021.
[53] X. Yue et al., "MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024; arXiv:2311.16502.
[54] C. E. Jimenez et al., "SWE-bench: Can language models resolve real-world GitHub issues?" in Proc. International Conference on Learning Representations (ICLR), 2024; arXiv:2310.06770.
[55] X. Yue et al., "MMMU-Pro: A more robust multi-discipline multimodal understanding benchmark," in Proc. ACL, 2025; arXiv:2409.02813.