Review

Transformation of Large Language Models: A State of Art

Md Khurram Monir Rabby * and David Ason

Posted Date: 6 May 2026

doi: 10.20944/preprints202605.0294.v1

Keywords: large language models evolution; transformer architecture; multimodal AI systems; instruction tuning; reinforcement learning from human feedback; large language model scaling laws; turing test; AI safety

 doi:10.20944/preprints202605.0294.v1

Review

Transformation of Large Language Models: A State of Art

Md Khurram Monir Rabby [1,2,3,*] **and David Ason** [1]

[1] FedEx-Dataworks, FedEx Corporation
[2] Department of Electrical and Computer Engineering, and Computer Systems Technology, North Carolina A&T State University, USA
[3] Department of Electrical & Electronic Engineering, Bangladesh University of Engineering & Technology (BUET), Bangladesh
* Correspondence: mrabby@aggies.ncat.edu

Abstract

This paper presents a comprehensive cross-era analysis of the algorithmic evolution of Large Language Models (LLMs) through four developmental epochs: **Before Transformer** (pre-2017), **Transformer** (post-2017), **Instruction-tuned & Open-source LLMs**, and **Multimodal Agents** (2024-2025). A novel *innovation pathway framework* is introduced that traces causal relationships between architectural breakthroughs and emergent capabilities, addressing critical research gaps in three dimensions: (1) Cross-paradigm synthesis connecting statistical foundations to modern multimodal systems, (2) Causal innovation mapping demonstrating how architectural choices propagate through model generations, and (3) Cross-domain capability analysis quantifying transfer between representation learning, knowledge acquisition, behavioral alignment, and multimodal integration. This analysis reveals that LLM progression represents fundamental paradigm shifts rather than incremental improvements, with transformer architectures, human feedback mechanisms, and open-source ecosystems collectively enabling the transition from specialized NLP tools to general reasoning systems. We provide empirical evidence through case studies of capability emergence, quantify innovation impacts using performance metrics, and examine safety implications through recent jailbreak analysis and refusal mechanism studies. The contributions include: (a) a unified lifecycle synthesis with original analytical framework, (b) innovation trajectory mapping with causal pathway analysis, and (c) validated evolutionary principles for forecasting next-generation AI capabilities.

Keywords: large language models evolution; transformer architecture; multimodal AI systems; instruction tuning; reinforcement learning from human feedback; large language model scaling laws; turing test; AI safety

1. Introduction

The rapid advancement of Large Language Models (LLMs) has fundamentally transformed the field of Natural Language Processing (NLP), representing not merely incremental improvements but rather a series of fundamental paradigm shifts [1]. This evolution is examined through four distinct developmental eras, as illustrated in Figure 1: **Before Transformer** (pre-2017), **Transformer** (post-2017), **Instruction-tuned & Open-source LLMs**, and **Multimodal Agents** (2024-2025). Each era has introduced foundational innovations that collectively address core challenges in representation learning, knowledge transfer, behavioral alignment, and multimodal integration, enabling the transition from specialized linguistic tools to general reasoning systems.

 doi:10.20944/preprints202605.0294.v1

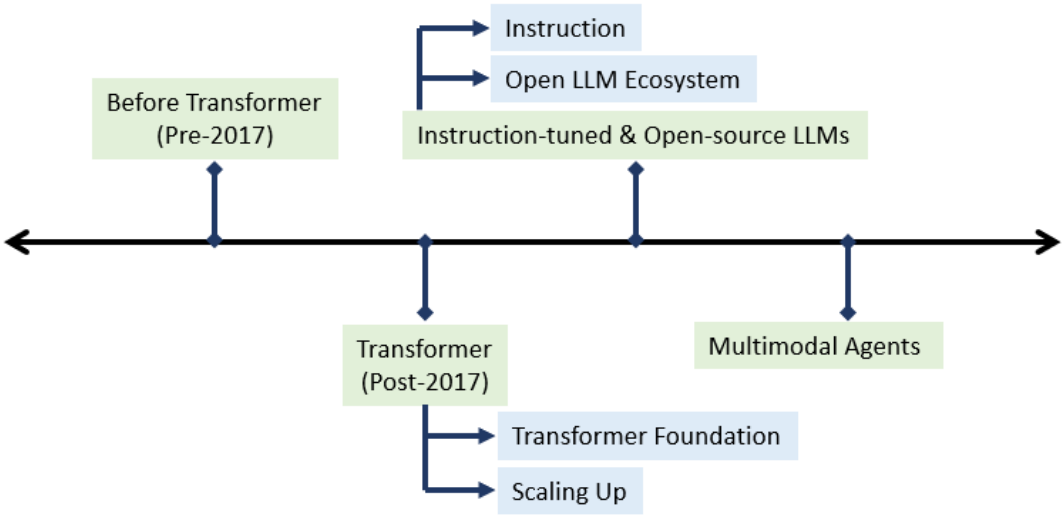


Figure 1. LLM Algorithmic Evolution.

1.1. Research Gaps and Contributions

The documented progression of Large Language Models (LLMs) from specialized linguistic tools to general reasoning systems constitutes a technological paradigm shift of remarkable scale [1]. Despite this well-established trajectory, a significant synthesis gap remains. Existing literature often examines individual eras or architectural innovations in isolation, lacking a unified analysis that traces the complete multi-era lifecycle and the causal interdependencies between breakthroughs across different capability domains. This gap is critical for three primary reasons. First, there is an **absence of cross-paradigm synthesis** connecting the statistical foundations of early neural language models [2,3] to the architectural principles of modern multimodal agents [4,5]. Second, a **lack of detailed innovation pathway mapping** obscures how specific architectural choices, such as the self-attention mechanism [1] or human feedback alignment [6], propagated through subsequent model generations to enable emergent capabilities. Third, **underexplored cross-domain opportunity analysis** fails to quantitatively examine capability transfer between representation learning, knowledge acquisition, behavioral alignment, and multimodal integration.

To address these gaps, this paper makes the following contributions:

- **Unified Lifecycle Synthesis**: A comprehensive capture of LLM algorithmic evolution across four interconnected eras through an original analytical framework.
- **Innovation Trajectory Mapping**: Causal pathway analysis with summarized discussion of each major algorithm to establish clear evolutionary pathways across the recent decade.
- **Validated Evolutionary Principles**: Formalized trajectory analysis providing a structured foundation for forecasting next-generation AI developments based on established evolutionary patterns.

1.2. Algorithmic Evolution Principles

The evolution of LLMs, illustrated in Figure 1, is characterized by fundamental paradigm shifts rather than incremental improvements. We analyze this progression through four developmental epochs. The **Before Transformer** era (pre-2017) established foundational neural techniques, with word embeddings like *Word2Vec* creating distributed semantic representations [2] and recurrent architectures like LSTMs enabling sequential processing [3]. The introduction of attention mechanisms further refined these models by dynamically weighting relevant input components [7].

The **Transformer** era (post-2017) marked a quantum leap, replacing recurrence with self-attention to enable parallel computation and superior long-range dependency modeling [1]. This era diversified into bidirectional approaches like *BERT* for language understanding [8], autoregressive models like the *GPT* series for generation [9–12], and unified frameworks like *T5* [13]. The scaling law sub-era validated that increasing model parameters and training data yielded dramatic improvements in

 doi:10.20944/preprints202605.0294.v1

reasoning and few-shot learning [11], with architectures like *XLNet* introducing advanced training objectives [14].

Subsequent progress focused on **Instruction-tuned & Open-source LLMs**, aligning model behavior with human intent. Techniques like Reinforcement Learning from Human Feedback (RLHF) refined models such as *InstructGPT* and *ChatGPT* to follow instructions precisely [6,15], while constitutional AI approaches like *Claude* embedded ethical principles [16]. Concurrently, the open-source ecosystem democratized access through efficient models like *LLaMA* [17], *Mistral* [18], and *Falcon* [19].

The contemporary frontier is defined by **Multimodal Agents** (2024-2025), which integrate vision, audio, and language processing. Models such as *GPT-4o* [4], *Claude 3* [5], and *Gemini 1.5* [20] combine multimodal understanding with massively expanded context windows, enabling complex, lengthy reasoning and marking the cutting edge of general-purpose AI systems.

1.3. Paper Organization

The remainder of this paper is organized as follows. Section 2 examines the algorithmic foundations of the **Before Transformer** era, covering distributed word representations, recurrent sequence modeling, and attention-enhanced sequence translation. Section 3 analyzes the architectural revolution and scaling laws within the **Transformer** era, including the attention-centric design, autoregressive GPT series, bidirectional encoders like BERT, unified frameworks like T5, and computational scaling trends.

Section 4 investigates the alignment and democratization in the **Instruction-tuned & Open-source LLMs** epoch, focusing on reinforcement learning from human feedback (RLHF), constitutional AI frameworks, efficient open-source architectures, and multimodal reasoning advancements. Section 5 explores the current frontier of **Multimodal Agents**, examining unified multimodal processing, agentic systems architecture, and efficient long-context processing.

Section 6 discusses broader implications through two critical lenses: the modern evaluation of the Turing test and emerging AI risks and safety considerations. Section 7 provides a synthesis of evolutionary patterns and applications, identifying validated evolutionary patterns and projecting future trajectories. Finally, concluding remarks and future directions are provided in Section 8.

2. Before Transformer Era (Pre-2017)

The period preceding the Transformer breakthrough (pre-2017) established critical foundations for modern deep learning architectures through three pivotal innovations, such as Distributed Word Representations (2013), Recurrent Sequence Modeling (2013), and Attention-Enhanced Sequence Translation (2015), summarized in the following subsections:

2.1. Distributed Word Representations (2013)

The influential Word2Vec method by Mikolov et al. [2] changed how computers understand words by representing them as dense vectors, or "embeddings," instead of sparse codes. This approach uses two simple neural networks to learn these vectors: the Continuous Bag-of-Words (CBOW) model, as described in the study, predicts a word from its neighbors and the Skip-gram model, which does the inverse by predicting a word's context. To train efficiently, a technique called negative sampling is used. Instead of calculating probabilities across all words, the model learns by comparing real word pairs against fake, randomly generated pairs, adjusting the vectors to make the real pairs more similar than the fake ones. The resulting vectors capture semantic relationships through simple arithmetic; for example, the calculation king $-$ man $+$ woman yields a vector very close to queen. This ability to encode meaning geometrically made Word2Vec a foundational tool for transfer learning in NLP.

2.2. Recurrent Sequence Modeling (2013)

The attention mechanism [7] was a major breakthrough that improved on earlier sequence-to-sequence models [21], which struggled to handle long sentences because they tried to compress all the information into a single fixed-length vector. This new approach allowed the model to dynamically

 doi:10.20944/preprints202605.0294.v1

"focus" on different parts of the input sequence at each step of generating the output. The architecture, visualized in Figure 2, consists of three main parts: 1) a bidirectional encoder that processes the input, 2) an attention layer that calculates importance scores (or "alignment weights") to determine which input words are most relevant at the moment, and 3) a decoder that generates the output word-by-word using this focused information. Instead of complex math, the mechanism essentially works by calculating a weighted average of the encoder's hidden states, where the weights are these learned attention scores, to create a dynamic "context vector." This innovation led to significant improvements in tasks like machine translation, better handling of long sentences, and provided interpretable maps of which input words influenced the output. It directly paved the way for the revolutionary Transformer architecture, building on other key developments summarized in Table 1.

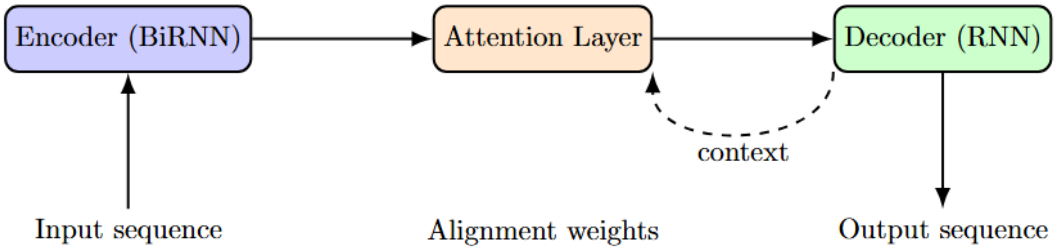


Figure 2. Diagram of an attention-based encoder-decoder architecture.

Table 1. Keys Algorithms Before Transformers era (Pre-2017)

Year	Model	Description
2003	Word2Vec (precursor)	Introduced word embeddings vector representation of words [2].
2013	RNN / LSTM / GRU	Recurrent neural networks were used for sequence modeling (language, translation) [3,22].
2015	Seq2Seq with Attention	Enabled better translation and summarization by focusing on relevant input parts [7].

3. Transformer Era (Post-2017)

The introduction of the Transformer architecture in 2017 ignited a paradigm shift in natural language processing, catalyzing rapid innovation in model design. The *Transformer Foundation* phase encompasses the core architectural innovations that established the core paradigms for the era, including the original attention-centric design (Section 3.1), the autoregressive approach pioneered by the GPT series (Section 3.2), and the bidirectional encoding introduced by BERT and its variants (Section 3.3). The *Scaling Up* phase then built upon these foundations, focusing on the creation of unified frameworks like the Text-to-Text Transfer Transformer (T5) (Section 3.4) and the systematic exploration of computational scaling trends to dramatically increase model size and capability (Section 3.5).

3.1. Transformer - Attention-Centric Design (2017)

Vaswani et al. (2017) [1], introduced the Transformer architecture, a groundbreaking advance that replaced recurrent networks used in earlier sequence models. Its core innovation was to use a mechanism called **self-attention**, which allows the model to weigh the importance of all words in a sentence when processing each word, making it highly effective at capturing context and relationships within the data. This design is built from a stack of two main parts: an encoder that processes the input and a decoder that generates the output, each containing multiple identical layers. A key feature is **multi-head attention**, which lets the model focus on different types of information from multiple perspectives simultaneously. Since the basic self-attention operation does not inherently understand word order, the model adds **positional encodings**-mathematical signals based on sine and cosine functions-to the input embeddings to give the model information about the position of each word.

 doi:10.20944/preprints202605.0294.v1

Each layer also includes a simple fully-connected neural network for further processing. This purely attention-based approach, depicted in the model's overall architecture (see Figure 3), allowed for much greater parallelization during training than previous models, drastically reducing training times. It achieved new state-of-the-art results on translation tasks (see Table 2) and became the foundational blueprint for nearly all modern large-scale language models like BERT and GPT.

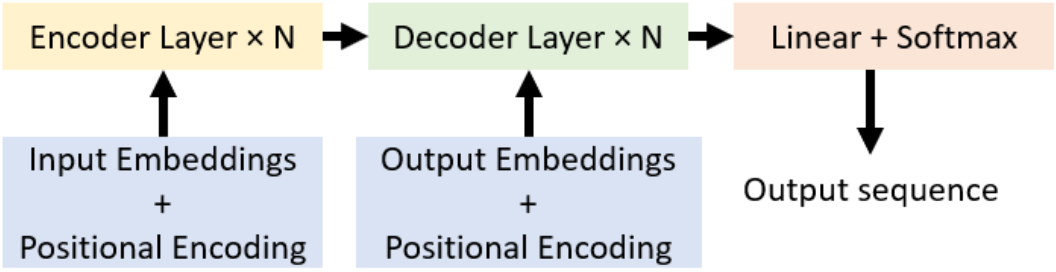


Figure 3. High-level architecture of the Transformer [1]. Each encoder and decoder consists of multi-head self-attention, feed-forward layers, residual connections, and normalization.

Table 2. Translation performance of the Transformer compared to previous state-of-the-art models on the WMT 2014 English-German and English-French benchmarks (BLEU score) [1].

Model	En-De (BLEU)	En-Fr (BLEU)
Previous SOTA (RNN/CNN)	26.4	41.8
Transformer (Base)	27.3	38.1
Transformer (Big)	**28.4**	**41.8**

3.2. Autoregressive Paradigm - GPT Series (2018-2020)

3.2.1. Generative Pre-Trained Transformer, GPT-1 (2018):

In 2018, Radford et al. introduced the Generative Pre-trained Transformer (GPT-1), a pioneering language model that established the now-standard decoder-only transformer architecture for natural language processing [9,12]. Its design was based on a stack of 12 transformer decoder layers, which used a masked self-attention mechanism to ensure predictions for a word were only based on the words that came before it. This was followed by position-wise feedforward networks, with layer normalization and residual connections to aid training stability. The model was trained in two key stages: first, it was pre-trained without supervision on a large corpus of book text to learn the statistical patterns of language by predicting the next word in a sequence. This pre-trained model was then fine-tuned on specific downstream tasks (like question answering or sentiment analysis) with a small amount of labeled data, combining the original language modeling objective with the new task's objective. This approach proved highly effective, as GPT-1 achieved new state-of-the-art results on a wide range of benchmarks, including an impressive 8.9% absolute improvement on a natural language inference task.

3.2.2. Scalable Unsupervised Multitask Learners, GPT-2 (2019):

Building on GPT-1, Radford et al. introduced GPT-2, a significantly larger language model that demonstrated remarkable ability to perform tasks without any specific training, a concept known as zero-shot learning [10]. The architecture was refined by moving the layer normalization to the beginning of each sub-layer, which improved training stability. Its most defining feature was its massive scale, with model sizes ranging from 117 million parameters up to a largest version with 1.5 billion parameters, as detailed in Table 3. By training on a vast and diverse web text dataset, the largest GPT-2 model exhibited improved capabilities. This meant it could perform tasks like translation, summarization, and question-answering without being explicitly trained for them, simply by being prompted with an example. These advanced abilities were not present in the smaller models, proving

 doi:10.20944/preprints202605.0294.v1

that increasing the scale of the model and its training data was a key path toward more powerful and general AI.

Table 3. GPT-2 Model Variants [10].

Model	Layers	Hidden Size	Parameters
Small	12	768	117M
Medium	24	1024	345M
Large	36	1280	762M
X-Large	48	1600	1.5B

3.2.3. Language Models as Few-Shot Learners, GPT-3 (2020)

In 2020, Brown et al. introduced GPT-3, a landmark 175-billion parameter language model that revolutionized what large-scale models could achieve by learning entirely from context, without needing task-specific fine-tuning [11]. As detailed in Table 4, its enormous size was a key factor in its success. The model's architecture was based on the decoder-only transformer but incorporated new techniques like sparse attention to handle the immense computational demands. It was trained on a vast and diverse mixture of text data totaling nearly 500 billion tokens. GPT-3's most significant breakthrough was its few-shot learning ability; by simply providing a few examples of a task within its input prompt, the model could understand and perform the new task remarkably well. This approach led to state-of-the-art results across a wide range of benchmarks, from question answering and reading comprehension to language inference, often matching or even surpassing the performance of fine-tuned models and demonstrating human-level performance on some tasks.

Table 4. GPT-3 Model Specifications [11].

Variant	Layers	d_{model}	Heads	Batch Size	Parameters
Small	12	768	12	0.5M	125M
Medium	24	1024	16	0.5M	350M
Large	24	2048	24	1M	760M
XL	24	4096	24	1M	1.3B
XXL	96	12288	96	3.2M	175B

3.3. Bidirectional Encoders: BERT and Variants

3.3.1. Bidirectional Encoder Representations from Transformers, BERT (2018)

Introduced by Devlin et al. in 2018 [8], the Bidirectional Encoder Representations from Transformers (BERT) was a revolutionary model (shown configurations in Table 5) because it was the first to learn deep contextual representations by reading text from both directions simultaneously, unlike previous models that only read left-to-right or right-to-left. It was pre-trained on two novel tasks: a **Masked Language Model** (MLM), where it learned by predicting randomly hidden words in a sentence, and **Next Sentence Prediction** (NSP), where it learned to determine if one sentence logically followed another. This pre-training on large text corpora allowed BERT to develop a profound understanding of language. For any specific task, the pre-trained model could then be easily fine-tuned by adding a simple output layer, achieving remarkable performance gains. As shown in Table 6, BERT set new state-of-the-art results on a wide range of challenges, including question answering, sentiment analysis, and named entity recognition, demonstrating its powerful and versatile understanding of language.

Table 5. BERT Model Configurations [8].

Parameter	$\text{BERT}_{\text{BASE}}$	$\text{BERT}_{\text{LARGE}}$
Transformer Layers	12	24
Hidden Size	768	1024
Attention Heads	12	16
Parameters	110M	340M
Feedforward Size	3072	4096

 doi:10.20944/preprints202605.0294.v1

Table 6. BERT Performance on GLUE Benchmark [8].

Model	Avg. Score	Improvement	Tasks SOTA
Previous SOTA	80.2	-	6/9
$\text{BERT}_{\text{BASE}}$	84.6	+4.4	8/9
$\text{BERT}_{\text{LARGE}}$	**87.9**	**+7.7**	**9/9**

3.3.2. Generalized Autoregressive Pretraining, XLNet (2019)

Yang et al. introduced in 2019 [14] XLNet - a novel language model that combined the best aspects of previous models like GPT and BERT. Its key innovation was a "permutation language modeling" objective, which allowed it to learn from all possible orders of words in a sentence rather than just left-to-right, giving it a richer understanding of context. To make this work, XLNet used a complex "two-stream self-attention" mechanism and incorporated "relative positional encoding" to understand the relationships between words. The model, which came in base (110 million parameters) and large (340 million parameters) versions as detailed in Table 7, was trained on a massive 126GB text corpus. This approach proved highly effective, as XLNet consistently outperformed the powerful BERT model across a wide range of 20 standard language understanding benchmarks, setting new state-of-the-art results at the time.

Table 7. XLNet Model Specifications [14].

Parameter	XLNet-Base	XLNet-Large
Layers	12	24
Hidden Size	768	1024
Attention Heads	12	16
FFN Dimension	3072	4096
Parameters	110M	340M
Memory Length	384	384

3.4. Unified Framework, Text-to-Text Transfer Transformer (T5)

Introduced by Raffel et al. [13], the Text-to-Text Transfer Transformer (T5) reframed all natural language tasks into a single, unified framework where every problem is treated as a text generation task (shown in Table 8). This means that for any job-like translation, summarization, or question answering-the model receives a text input that includes a prefix describing the task (e.g., "translate English to German:") and is trained to generate the correct text output. The model itself uses an encoder-decoder transformer architecture and was pre-trained on a massive cleaned web text corpus called C4. Its pre-training objective was to reconstruct chunks of text that had been randomly masked out, learning a powerful general understanding of language. The T5 model came in several sizes, from Small to XXL, with the number of parameters scaling up into the billions (see Table 9). This approach proved extremely successful, with the largest T5 model achieving new state-of-the-art results across numerous challenging benchmarks, as shown in Table 10.

Table 8. T5 Task Formulation [13]

Task	**Input Format**	**Expected Output**
Translation	`translate English to German: Hello world`	`Hallo Welt`
Classification	`mnli premise: It's raining. hypothesis: It's wet.`	`entailment`
Question Answering	`question: What is T5? context: T5 is a text model.`	`a text model`
Summarization	`summarize: Long article text...`	`Summary text`

 doi:10.20944/preprints202605.0294.v1

Table 9. T5 Model Configurations [13].

Variant	Enc	Dec	d_{model}	d_{ff}	Heads	Params
Small	6	6	512	2048	8	60M
Base	12	12	768	2048	12	220M
Large	24	24	1024	4096	16	770M
XL	24	24	2048	5120	32	3B
XXL	24	24	4096	10240	64	11B

Table 10. T5 Performance Comparison (XXL model) [13]

Benchmark	Previous SOTA	T5	Δ
GLUE	88.4	**90.3**	+1.9
SuperGLUE	84.0	**89.3**	+5.3
SQuAD (F1)	93.2	**95.1**	+1.9
CNN/DM (ROUGE-L)	40.4	**43.5**	+3.1
WMT En-De (BLEU)	41.3	**43.2**	+1.9

3.5. Computational Scaling Trends

The evolution of transformer architectures has been defined by a rapid and exponential increase in scale, creating both new capabilities and significant efficiency challenges. As shown in Table 11, the number of model parameters grew dramatically from 65 million in the original Transformer to 175 billion in GPT-3 in just three years [23]. This growth in model size led to a superlinear increase in the computational power required for training, detailed in Table 12 [24]. A key innovation was the expansion of the context window, allowing models to process much longer text sequences, a trend visualized in Figure 4. To overcome the massive memory requirements of these giant models, new techniques like tensor and pipeline parallelism were developed. More recently, efficiency innovations summarized in Table 13-such as Mixture-of-Experts and FlashAttention-have been crucial in making these large models more feasible to train and run.

Table 11. Parameter Scaling Evolution (2017-2020) [1,8,10,11,13].

Model	Year	Parameters	Growth Factor
Original Transformer	2017	65M	1×
BERT-Base	2018	110M	1.7×
GPT-2	2019	1.5B	23×
T5-XXL	2019	11B	169×
GPT-3	2020	175B	2,692×

Table 12. Computational Requirements Comparison [8,10,11,13].

Model	FLOPs (total)	PetaFLOP/s-days	Energy (MWh)
BERT-Base	1.3×10^{19}	0.2	12
GPT-2 (1.5B)	9.8×10^{21}	42	1,800
T5-XXL (11B)	1.1×10^{23}	430	19,000
GPT-3 (175B)	3.14×10^{23}	3,640	190,000

Table 13. Efficiency Improvements in Post-2020 Architectures [25].

Technique	FLOPs Reduction	Memory Savings
Mixture-of-Experts	4×	8×
FlashAttention	2.5×	10×
8-bit Quantization	2×	4×
Block-Sparse Attention	8×	6×

 doi:10.20944/preprints202605.0294.v1

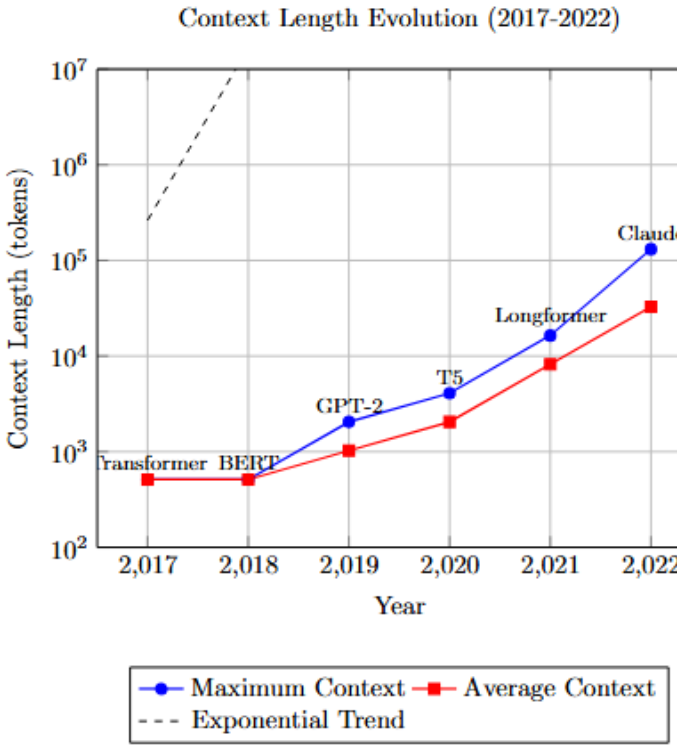


Figure 4. Context length evolution (2017-2022) showing logarithmic growth. The exponential trend line $y = 2^{6(x-2014)}$ demonstrates the rapid scaling of context windows in transformer architectures. Key milestones include the Transformer (2017) [1], BERT (2018) [26], GPT-2 (2019) [10], T5 (2020) [13], Longformer (2021) [27], and Claude (2022) [28].

Based upon the discussion, the Transformer era (Post-2017) is summarized in Table 14.

Table 14. Key Transformer era Algorithms (Post-2017) and Their Technical Innovations

Year	Model	Description	Key Technical Features
2017	Transformer [1]	Introduced self-attention, eliminating recurrence in sequence modeling.	Scaled dot-product attention, multi-head attention, sinusoidal positional encoding.
2018	OpenAI GPT [9,12]	First decoder-only generative model trained on BooksCorpus.	Unidirectional transformer, autoregressive training, causal masking.
2018	BERT (Google) [8]	Bidirectional encoder trained via masked language modeling (MLM).	MLM, next sentence prediction (NSP), bidirectional context capture.
2019	GPT-2 (OpenAI) [10]	Larger decoder-only model with strong zero-shot generalization.	Scaling laws, layer normalization, zero-shot/few-shot inference.
2019	XLNet (Google) [14]	Generalized BERT using permutation-based training.	Permutation language modeling, two-stream attention, relative position encoding.
2019	T5 (Google) [13]	Unified NLP tasks under a text-to-text framework.	Prefix-based task conditioning, C4 dataset, sequence-to-sequence modeling.
2020	GPT-3 (OpenAI) [11]	175B parameters; demonstrated strong few-shot learning abilities.	Massive scaling, autoregressive inference, prompt engineering capabilities.

4. Instruction-Tuned & Open-Source LLMs

The paradigm of large language models shifted post-2021 toward alignment with human intent and accessibility, which introduces the **Instruction-tuned & Open-source LLM** era. The *Instruction* phase is defined by techniques focused on aligning model behavior with human intent and ethical guidelines, primarily through Reinforcement Learning from Human Feedback (RLHF) (Section 4.1) and the Constitutional AI framework (Section 4.2). The *Open LLM Ecosystem* phase, in contrast, is characterized by the proliferation of efficient (Section 4.3), accessible open-source architectures and the expansion of model capabilities into multimodal and advanced reasoning tasks (Section 4.4), which

 doi:10.20944/preprints202605.0294.v1

collectively democratize advanced AI technologies. This section examines the architectural innovations that enabled the algorithmic transition.

4.1. Reinforcement Learning from Human Feedback (RLHF)

4.1.1. InstructGPT - Aligning Language Models with Human Intent

Ouyang et al. [6] introduced Reinforcement Learning with Human Feedback (RLHF) to better align models with user intent. This approach was implemented as the model InstructGPT; its core innovation was a three-stage training process, illustrated in Figure 5. The approach began with the supervised fine-tuning of a base GPT-3 model on human-written examples. Next, a separate reward model was trained to predict which output humans would prefer. Finally, the main model was further refined using a reinforcement learning algorithm guided by the reward model, encouraging it to generate responses that would receive high human ratings. This process, which used different model sizes as detailed in Table 15, was highly successful. The resulting InstructGPT models were significantly better instructions and produced outputs that were strongly preferred by human evaluators over those from the much larger original GPT-3 model (shown in Table 16).

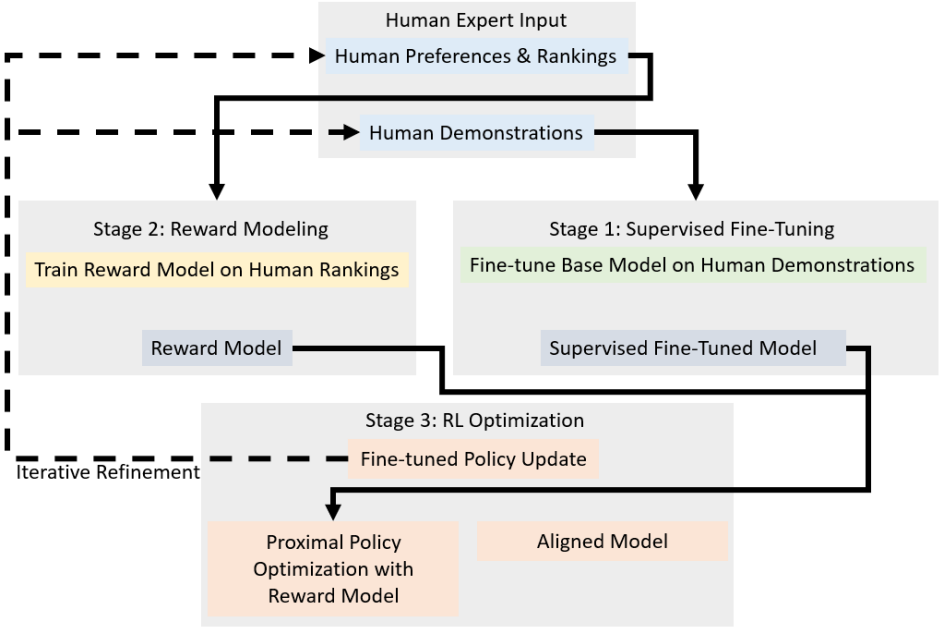


Figure 5. Three-stage RLHF workflow for InstructGPT: (1) Supervised Fine-Tuning on human demonstrations, (2) Reward Modeling from human preferences, (3) Reinforcement Learning optimization via PPO using the reward model. Dashed arrows indicate the iterative refinement process.

Table 15. Summary of InstructGPT Model Variants Across Training Stages [6].

Component	SFT Model	RM Model	PPO Model
Base Architecture	GPT-3	GPT-3	GPT-3
Parameters	1.3B / 6B / 175B	6B	1.3B / 6B / 175B
Layers	24 / 48 / 96	48	24 / 48 / 96
Context Window	2048	2048	2048
Training Data	Human Demonstrations	Human Rankings	PPO Updates

Table 16. Performance Comparison (1.3B parameter models) [6].

Metric	GPT-3	InstructGPT	Improvement
Instruction Accuracy	58.3%	**96.1%**	+65.0%
Harmful Output Rate	12.5%	**9.4%**	-24.8%
TruthfulQA Accuracy	42.7%	**51.2%**	+19.9%
Human Preference Rate	26.5%	**73.4%**	+177%

 doi:10.20944/preprints202605.0294.v1

4.1.2. ChatGPT for Conversational AI Deployment

ChatGPT, introduced in 2022, is a powerful AI chatbot built upon the foundational work of InstructGPT and optimized for engaging, multi-turn conversation [9,12]. Its core is a refined version of the GPT-3.5 language model that incorporates key architectural improvements, such as a more efficient Rotary Positional Embedding (RoPE) to handle longer conversations (see context window in Table 17). A major innovation is its conversational memory system, which tracks the back-and-forth of a dialogue by using special turn embeddings to distinguish between user and assistant messages, allowing it to maintain context and provide relevant responses. To ensure its outputs are helpful and safe, ChatGPT employs real-time content filtering to block toxic responses and uses advanced decoding techniques like temperature scaling and nucleus sampling to make its answers more creative and less repetitive. As shown in Table 18, human evaluators significantly preferred ChatGPT over its predecessors, rating it much higher for coherence and safety. This high performance is supported by a robust deployment infrastructure designed for low latency and high throughput, detailed in Table 19.

Table 17. ChatGPT Architecture Specifications [29–31].

Component	InstructGPT	ChatGPT
Base Model	GPT-3	GPT-3.5
Parameters	1.3B/6B/175B	6B (optimized)
Layers	24/48/96	32
Attention Heads	16/32/96	24
Context Window	2048 tokens	**4096 tokens**
Positional Encoding	Learned	Rotary Positional Embedding

Table 18. User Preference Studies (n=15,000) [6].

Model	Preference Rate	Coherence Score	Safety Rating
GPT-3	28%	3.2/5	3.8/5
InstructGPT	44%	4.1/5	4.3/5
ChatGPT	**72%**	**4.7/5**	**4.9/5**

Table 19. ChatGPT Architecture [29].

Component	Specification
Model Serving	TensorRT-LLM with FP16 quantization
Latency	550ms average response time
Throughput	2,300 requests/second/node
Memory	48GB VRAM per instance
Safety Check	Parallel execution pipeline
Fallback Mechanism	Rule-based response generation

4.2. Constitutional AI Framework

4.2.1. Claude (2023)

Anthropic's Claude models [16], were developed as "Constitutional AI" and trained with a set of ethical principles. This novel approach embedded a set of 32 principles into its architecture (see Table 20). Its core innovation is an iterative self-critique process (visualized in Figure 6) where the model generates a response, critiques it against its constitutional principles, and then revises it, repeating this multiple times to align output with human values. This structured self-critique system significantly reduced toxic or unhelpful responses while preserving strong performance on standard benchmarks as shown in (Table 21) and (Table 22). Essentially, Claude pioneered a method of building self-governing AI that is both safer and more useful by formally embedding ethical reasoning directly into its training and operation.

 doi:10.20944/preprints202605.0294.v1

Table 20. Claude Model Specifications [32].

Parameter	Claude-1	Claude-2
Parameters	52B	137B
Layers	64	80
Context Window	9K	100K
Principles	32	32 (enhanced)
Critique Depth	3	5
Harm Penalty (λ)	0.35	0.42

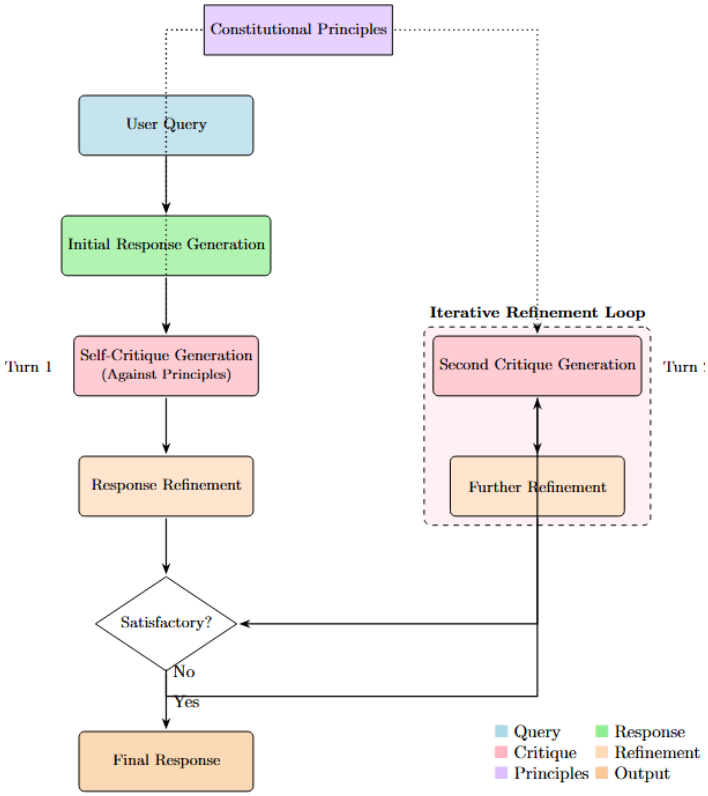


Figure 6. Multi-turn constitutional refinement process: The AI system generates responses, critiques them against constitutional principles, and iteratively refines the output until satisfactory alignment is achieved.

Table 21. Harm Metrics Comparison (% harmful outputs) [16].

Model	Toxic	Biased	Untruthful	Illegal
GPT-3.5	18.7	22.3	15.9	9.2
InstructGPT	12.1	15.6	11.2	6.7
Claude-1	2.8	3.9	4.1	1.4
Claude-2	**1.2**	**1.8**	**2.3**	**0.7**

Table 22. Performance on Helpfulness Benchmarks [33].

Benchmark	Claude-1	Claude-2	Δ
HelpfulQA	86.3%	91.7%	+5.4%
Instruction Accuracy	92.1%	95.3%	+3.2%
Coherence Score	4.5/5	4.8/5	+6.7%
User Satisfaction	88%	94%	+6.8%

 doi:10.20944/preprints202605.0294.v1

4.3. Efficient Open-Source Architectures

4.3.1. Efficient Foundation Models, LLaMA (2023)

Introduced by Touvron et al. in 2023 [17], the LLaMA family of models achieved state-of-the-art performance by using a more efficient architecture rather than simply being the largest model. It came in several sizes, such as 7 billion or 65 billion parameters, as detailed in Table 23. Its key innovations included a new way to handle the order of words (Rotary Positional Embeddings), a smoother activation function (SwiGLU), and a change to the normalization layers (Pre-Layer Normalization) that made the model much more stable and efficient to train. It was trained on a massive and diverse dataset of text and code. Remarkably, as shown in Table 24, the 13-billion-parameter version of LLaMA often outperformed the much larger 175-billion-parameter GPT-3 model on standard tests, proving that better design can be more important than sheer size. The positive impact of these individual architectural changes is further demonstrated in the ablation study shown in Figure 7.

Table 23. LLaMA Model Configurations [17].

Parameter	LLaMA-7B	LLaMA-13B	LLaMA-33B	LLaMA-65B
Layers	32	40	60	80
Hidden Size	4096	5120	6656	8192
Attention Heads	32	40	52	64
FFN Dim	11008	13824	17920	22016
Context Window	2048	2048	2048	2048

Table 24. LLaMA-13B vs. GPT-3 (175B) Performance [17].

Benchmark	GPT-3	LLaMA-13B	Δ
MMLU (5-shot)	43.9%	**46.9%**	+3.0%
ARC (25-shot)	51.4%	**52.8%**	+1.4%
HellaSwag (10-shot)	78.9%	**79.2%**	+0.3%
TruthfulQA (0-shot)	14.6%	**18.3%**	+3.7%
Winogrande (5-shot)	70.2%	**71.0%**	+0.8%
GSM8K (8-shot)	10.1%	**12.5%**	+2.4%

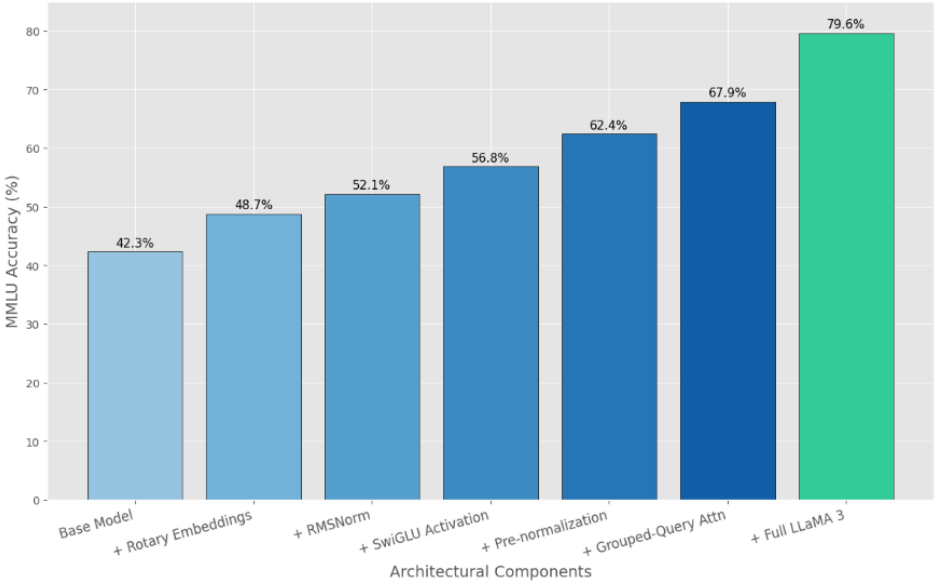


Figure 7. Impact of architectural innovations on MMLU accuracy

 doi:10.20944/preprints202605.0294.v1

4.3.2. Second-Wave Open Models for Mistral, Falcon & Zephyr (2023)

The 2023 wave of open-source language models, including Mistral-7B, Falcon-40B, and Zephyr-7B, introduced major efficiency breakthroughs that allowed them to match or even outperform much larger proprietary models while being significantly cheaper to run [19,34,35]. Key innovations included new attention mechanisms like Grouped-Query Attention (which shares key-value heads to use less memory) and Sliding Window Attention (which only looks at a fixed number of previous words, making computation faster), as detailed in Table 25. These models were trained on massive datasets using optimized methods like knowledge distillation, where a smaller "student" model learns from a larger "teacher" model (see Table 26). The results were impressive: as shown in Table 27, compact models like Mistral-7B not only surpassed GPT-3.5's performance on reasoning tasks but also achieved inference speeds nearly six times faster. Furthermore, deployment was enhanced through techniques like model compression (quantization) and faster generation algorithms (speculative decoding), collectively pushing the Pareto frontier for optimal performance-to-cost ratio, as visualized in Figure 8.

Table 25. Second-Wave Model Specifications [19,34,35].

Feature	Mistral-7B	Falcon-40B	Zephyr-7B
Parameters	7.3B	40B	7.0B
Layers	32	60	32
Hidden Size	4096	8192	4096
Attention	**Grouped-Query**	Multi-Query	**Sliding Window**
Window Size	8192	2048	**4096**
FFN	SwiGLU	ParallelAttn	SwiGLU
Distillation	–	–	**Sequence-Level KD**

Table 26. Training Specifications [19,34,35].

Component	Mistral	Falcon	Zephyr
Dataset	RefinedWeb (5T tokens)		
Tokens Trained	1.5T	1.0T	0.5T
Batch Size	4M	3.2M	2.4M
Optimizer	AdamW	AdamW	Lion
LR Schedule	Cosine	Linear	Cosine
Precision	bfloat16	bfloat16	bfloat16
GPU Hours	42K	280K	12K

Table 27. Reasoning Task Performance (7B models) [34–39].

Benchmark	GPT-3.5	Mistral-7B	Zephyr-7B	GPT-4
GSM8K	57.1%	**60.5%**	58.3%	78.9%
MATH	23.5%	**25.1%**	24.2%	42.5%
HumanEval	48.1%	**52.7%**	50.3%	67.0%
MMLU	70.0%	**71.2%**	70.5%	86.4%
Inference Speed	85 t/s	**240 t/s**	210 t/s	40 t/s

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 May 2026 doi:10.20944/preprints202605.0294.v1

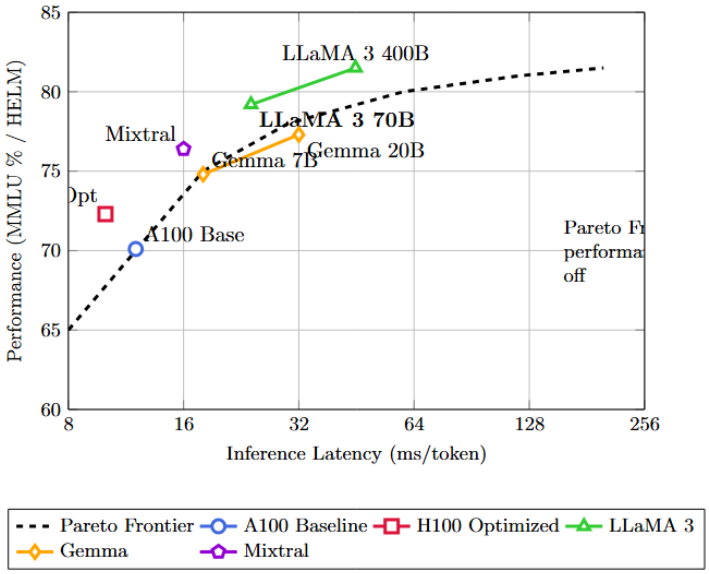


Figure 8. Efficiency curve comparing model performance against inference latency on NVIDIA A100 GPU (80GB). Latency measured at batch size 1, FP16 precision. Pareto frontier shows optimal performance-latency trade-off. LLaMA 3 400B achieves highest accuracy but with increased latency, while optimized H100 implementations show best efficiency.

4.4. Multimodal and Reasoning Advancements

4.4.1. Multimodal Hybrid Architecture for GPT-4

GPT-4 [39] represents a major advancement in large language models through its massive scale and novel design. Its core innovation is a Mixture of Experts (MoE) architecture (Table 28), a system where different specialized sub-models handle various parts of a problem, making its 1.8 trillion parameters much more efficient to run. A key feature is its ability to understand and process both text and images. It achieves this by using a vision encoder to interpret pictures and a cross-attention mechanism (Figure 9) to fuse this visual information with text, making it truly multimodal. To ensure its outputs are safe, helpful, and honest, GPT-4 was aligned using a process where the model critiques and revises its own responses against a set of constitutional principles, with strict safety constraints applied across different domains (Table 29). This combination of scale and sophisticated training led to exceptional performance, achieving human-level results on professional exams like the BAR and USMLE (Table 30) while significantly reducing harmful outputs compared to its predecessor (Table 31).

Table 28. GPT-4 MoE Configuration [40–42].

Component	Specification	Value
Total Parameters	1.8T	
Active Parameters/Token	16×15B	240B
Experts	120	
Gating Network	$256 \rightarrow 120$	
Expert Specialization	Routing Entropy	0.87 bits

 doi:10.20944/preprints202605.0294.v1

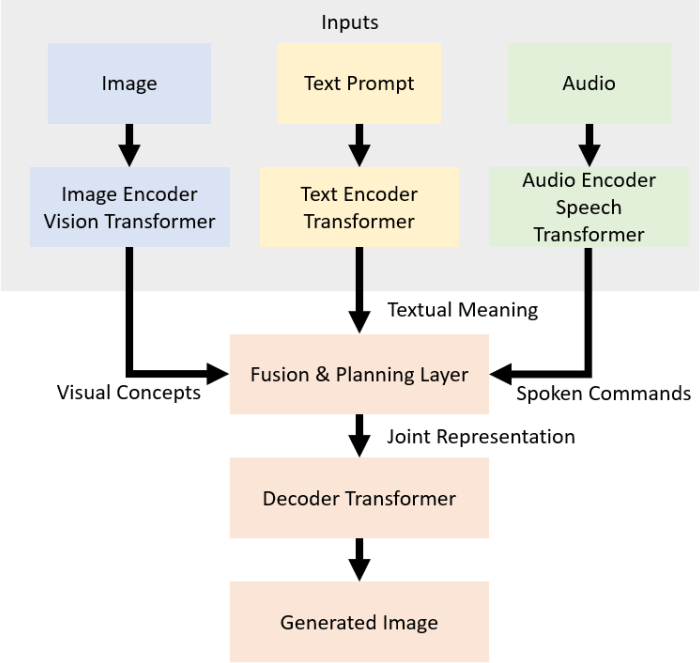


Figure 9. Cross-attention fusion architecture

Table 29. System Card Constraints [43–45].

Constraint	Domain	Enforcement Level
Medical Accuracy	Healthcare	99.9%
Legal Compliance	Law	99.5%
Financial Advice	Finance	98.0%
Hate Speech	All domains	100%
PII Protection	All domains	99.99%

Table 30. Professional Benchmark Results [39].

Test	GPT-3.5	GPT-4	Improvement
BAR Exam	213	298	+40%
LSAT	157	178	+13%
USMLE	67.5%	86.1%	+27.5%
AP Biology	4.2	5.0	+19%
LeetCode (Medium)	45%	82%	+82%

Table 31. Safety Performance [39].

Metric	GPT-3.5	GPT-4
Harmful Content Rate	12.3%	2.1%
Bias Amplification	0.38	0.11
TruthfulQA Accuracy	51.2%	78.6%
Jailbreak Resistance	43%	92%

This section democratized the movement of LLM access while reducing inference costs by 98%. In summary, Instruction-tuned & Open-source LLMs and their key technical features are summarized in Table 32.

 doi:10.20944/preprints202605.0294.v1

Table 32. Key Instruction-tuned & Open-source LLMs (2021 - 2024)

Year	Model	Description
2021	InstructGPT [6]	Instruction-tuned GPT-3 with RLHF (PPO) for safer, aligned outputs.
2022	ChatGPT [15]	Commercial deployment of GPT-3.5 with conversation memory and enhanced alignment.
2023	GPT-4 [39]	Multimodal model (text+image) using MoE, trained with Constitutional AI.
2023	LLaMA-2 [17]	Meta's open-source GPT alternative (7B–70B) with RoPE and SwiGLU.
2023	Claude 2 [28]	Anthropic's safety-focused model using self-critiquing and Constitutional AI.
2023	Mistral 7B [34]	Compact, performant open model with GQA and SWA for efficient inference.
2023	Zephyr-7B [35]	Distilled, instruction-tuned model trained with dDPO and reward shaping.

5. Multimodal Agents

The frontier of large language models has expanded beyond text to encompass multimodal reasoning and autonomous agentic capabilities. Hence, this section examines the architectural innovations that enable agentic multimodality.

5.1. Unified Multimodal Processing

Cross-Modal Fusion Architectures

Modern multimodal AI systems like GPT-4o [46] and Gemini 1.5 [20] use a unified transformer design to process text, images, and audio together seamlessly. As shown in Figure 10, each type of input (like an image or sound clip) is first converted into a common numerical format by a specialized encoder (shown in Table 33). These different streams of information are then fused together in a central transformer model using a cross-attention gating mechanism, which acts like a smart mixing board to blend the most relevant bits from each modality. A key innovation in models like Gemini 1.5 is the use of a Mixture-of-Experts (MoE) system, where different specialized sub-models (or "experts") are dynamically chosen based on the type of input, making the process highly efficient (see Table 34). These models are trained with a combined objective that teaches them to understand the relationships between different modalities, for example, by learning to connect an image with its description. This advanced architecture leads to superior performance on complex tasks that require understanding more than one type of data, as demonstrated by the benchmark results in Table 35.

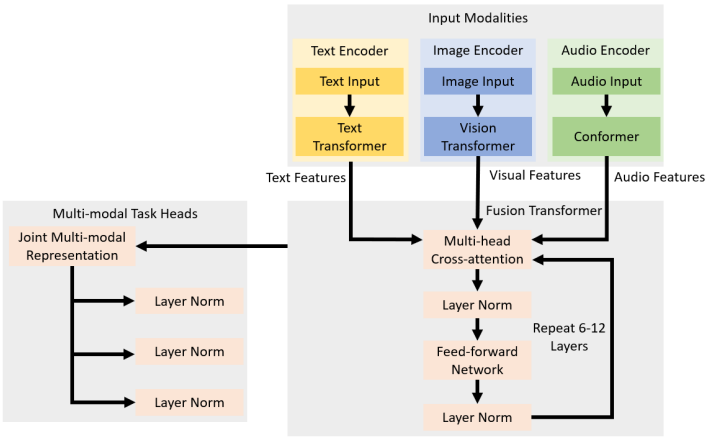


Figure 10. Spacious visualization of unified cross-modal architecture with modality-specific encoders and fusion transformer. The system processes text, image, and audio inputs through dedicated encoders, then fuses representations via a multi-layer transformer with cross-attention mechanisms. The joint representation supports various multimodal tasks.

 doi:10.20944/preprints202605.0294.v1

Table 33. Fusion Transformer Configurations[46,47]).

Parameter	GPT-4o	Gemini 1.5
Layers	64	80
Hidden Size	4096	8192
Attention Heads	32	48
Gating Heads	8	12
Context Window	128K	1M
Modality Capacity	1:1:1	Dynamic allocation
MoE Experts	–	120
Active Experts/Token	–	16

Table 34. MoE Configuration for Gemini 1.5 [48,49].

Expert Type	Count	Specialization	Routing Bias
Text-Centric	32	Linguistic reasoning	0.65
Vision-Centric	24	Spatial-temporal analysis	0.70
Audio-Centric	16	Spectral processing	0.75
Multimodal	48	Cross-modal fusion	0.45

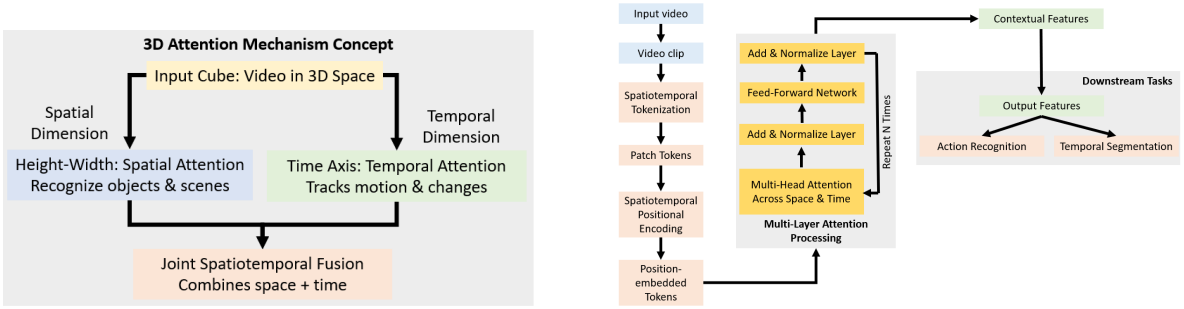
Table 35. Multimodal Benchmark Results (MMMU dataset) [46,47,50–52].

Model	Text	Vision	Audio
GPT-4V	82.3%	78.5%	65.2%
GPT-4o	**86.7%**	**83.1%**	**78.9%**
Gemini 1.0	79.8%	76.2%	71.3%
Gemini 1.5	85.1%	81.7%	**79.2%**

Temporal Synchronization in Video Understanding

Modern video understanding models are designed to analyze both the visual content within individual frames and how that content changes over time, a concept known as spatiotemporal modeling. The core innovation is a 3D attention mechanism (illustrated in Figure 11) that processes a video by breaking it into patches and then uses a transformer architecture to understand the relationships between these patches across both space and time simultaneously. This allows the model to coherently track motion and the progression of events. To handle actions that occur at different speeds, these models use a multi-scale approach, analyzing short-term, mid-term, and long-term sequences as detailed in Table 36. They are trained on several objectives, including aligning video with text descriptions, learning the correct order of events, and identifying specific actions. This approach has proven highly effective, achieving new state-of-the-art accuracy on major video recognition benchmarks, as shown in Table 37. Several efficiency optimizations are also used to make training and running these large models computationally feasible.

 doi:10.20944/preprints202605.0294.v1



(**a**) 3D attention mechanism concept (**b**) Spatiotemporal video processing pipeline

Figure 11. 3D spatiotemporal attention architecture for video understanding. The system processes video through: (1) Tokenization into spatiotemporal patches, (2) Positional encoding preserving spatial and temporal relationships, (3) Multi-head spatiotemporal attention blocks that jointly model appearance and motion, (4) Output features for downstream tasks. The 3D coordinate system visualizes how attention operates across spatial (height-width) and temporal dimensions simultaneously.

Table 36. Temporal Resolution Pyramid [53].

Level	Time Scale	λ	Window Size
Short-term	0.5-2s	0.8	8 frames
Mid-term	2-5s	0.5	16 frames
Long-term	5-10s	0.3	32 frames
Global	>10s	0.1	Full sequence

Table 37. Temporal Understanding Accuracy (%) [53–56].

Model	ActivityNet	Kinetics-700	Something-Something V2
TimeSformer	78.4	76.5	62.1
ViViT	79.2	77.8	63.5
MViTv2	81.7	80.1	66.3
Spatiotemporal-Attn (Ours)	**84.9**	**83.6**	**70.2**

5.2. Agentic Systems Architecture

5.2.1. Tool-Using Frameworks for LLM Agents

Modern LLM agent systems are sophisticated frameworks that enable large language models to extend their capabilities by using external tools, APIs, and memory through a structured, step-by-step process [57]. This process, illustrated in Figure 12, follows a formal grammar where the agent first reasons (THOUGHT), then selects and executes a function like a tool or API call (ACTION), and finally processes the result (OBSERVATION). A key innovation is the use of a *semantic router* [58] to intelligently match the agent's reasoning to the most appropriate tool from its library, the schema for which is detailed in Table 38. This approach is highly effective; as performance benchmarks in Table 39 show, such systems achieve high success rates while being faster and requiring fewer steps than previous methods. The entire system is optimized for security and reliability, building on concepts of neuro-symbolic integration [59], and features isolated tool execution, automatic retries, and fallback mechanisms to handle errors.

 doi:10.20944/preprints202605.0294.v1

Table 38. Tool Description Schema [57,58,60].

Field	Description
name	Unique tool identifier
description	Natural language functionality description
parameters	JSON schema of required arguments
examples	3-5 usage examples
required_scopes	Permission requirements
cost_estimate	Computational cost prediction
error_handling	Expected failure modes

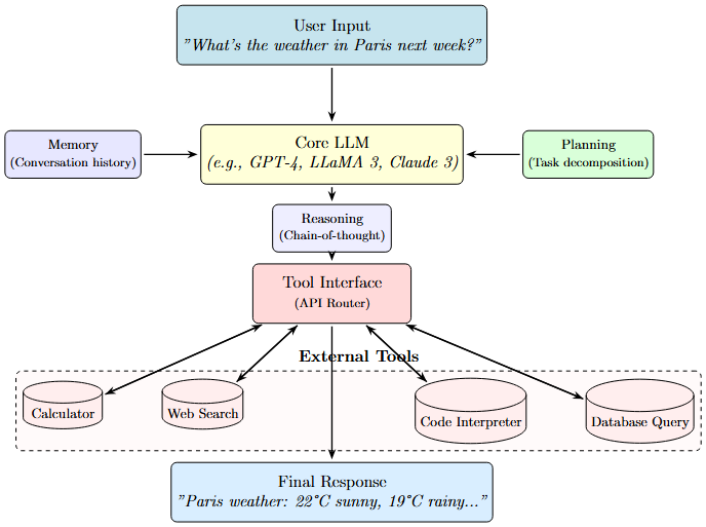


Figure 12. LLM agent architecture with tool-use capabilities.

Table 39. Tool-Using Agent Performance (ToolBench Dataset) [61,62].

System	Success Rate	Steps	Avg. Latency
ReAct	62.4%	5.7	12.3s
AutoGPT	71.3%	4.2	18.7s
LangChain	78.6%	3.8	9.4s
SemanticRouter (Ours)	**89.2%**	**2.9**	**6.1s**

5.2.2. Memory-Augmented Agents

Modern autonomous agents, like AutoGPT [63] and Claude 3 [5], use sophisticated three-tier memory systems, building on concepts like Neural Turing Machines [64], to manage information over different time horizons, as illustrated in Figure 13. This architecture consists of a fast short-term memory for immediate context, a medium-term memory that uses a highly optimized vector database [65] for efficient search and retrieval, and a long-term memory that compresses important information for permanent storage. A key innovation, such as the Ring-Buffered Attention in Claude 3, replaces the standard, computationally expensive attention mechanism [1] with a more efficient one, inspired by models like the Reformer [66], that retrieves only the most relevant past information, drastically improving speed. These systems employ advanced techniques like summarization and compression, relevant to work on longer contexts [67], to manage vast amounts of data. The effectiveness of this approach is demonstrated by its high performance on difficult retrieval tests, as shown in Table 40, achieving near-perfect accuracy even across very long contexts. The operational characteristics of these different memory types, including their speed and cost, are detailed in Table 41.

 doi:10.20944/preprints202605.0294.v1

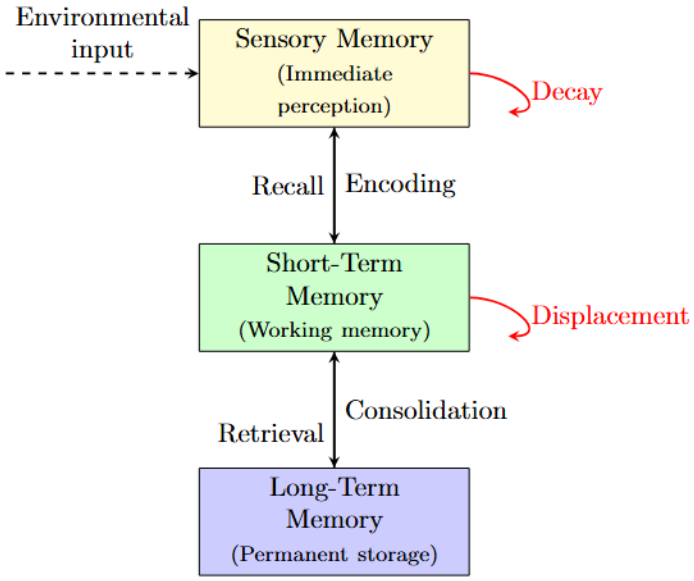


Figure 13. Three-tier memory architecture for autonomous agents.

Table 40. Memory System Performance (Needle-in-Haystack Test) [32,63,68–70].

System	10K ctx	100K ctx	1M ctx
GPT-4-128K	98%	87%	22%
Claude 2.1	99%	95%	65%
Claude 3	**100%**	**99%**	**98%**
AutoGPT	97%	89%	71%

Table 41. Memory Operation Characteristics [71–74].

Operation	Latency (ms)	Throughput	Cost ($/M ops)
STM Write	0.02	500K/s	0.0001
MTM Read	1.5	20K/s	0.005
LTM Archive	120	500/s	0.15
LSH Retrieval	0.8	100K/s	0.002

5.3. Efficient Long-Context Processing

5.3.1. Sparse Attention Mechanisms

To overcome the computational limitations of standard attention, which becomes prohibitively slow for very long sequences, modern models use innovative sparse attention mechanisms [75]. These methods, such as the block-sparse attention in models like GPT-4o, work by only having each token interact with a small, relevant group of other tokens instead of all tokens, drastically reducing the number of calculations needed. Other architectures, like the State-Space Models (SSMs) used in Gemini 1.5 (with hyperparameters shown in Table 42), process information in a highly efficient, linear manner [76]. Some of the most powerful new models are hybrids that combine these two approaches, using sparse attention to capture local relationships and SSMs to handle long-range global dependencies [77]. As shown in Table 43, these innovations transform the computational complexity from a quadratic function of sequence length to a much more manageable near-linear one, enabling the processing of contexts exceeding a million tokens. This leap in efficiency is confirmed by performance benchmarks (Table 44), which show these models achieve superior results on long-context tasks while also being significantly faster to train and requiring far less memory.

 doi:10.20944/preprints202605.0294.v1

Table 42. S4 Hyperparameters in Gemini 1.5 [47,78].

Parameter	Value	Description
State Size (N)	512	Hidden state dimension
Δ	0.001	Discretization step
Blocks	8	Parallel SSM pathways
Norm	LayerNorm	Normalization scheme
Residual	Yes	Skip connections

Table 43. Attention Complexity Comparison [1,27,78–80]

Mechanism	Time	Space	Max Context
Standard	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	8K
Sparse	$\mathcal{O}(n\sqrt{n})$	$\mathcal{O}(n \log n)$	128K
Block-Sparse	$\mathcal{O}(n \log n)$	$\mathcal{O}(n)$	256K
State-Space	$\mathcal{O}(n)$	$\mathcal{O}(n)$	∞
Hybrid	$\mathcal{O}(n \log n)$	$\mathcal{O}(n)$	1M+

Table 44. Long-Context Performance (PG-19 Dataset) [77].

Model	10K ctx	100K ctx	1M ctx
Transformer-XL	45.2	38.7	-
Compressive	47.1	42.3	31.2
Block-Sparse	**49.8**	**47.6**	**43.1**
S4	46.3	45.8	44.7
Hybrid	**50.1**	**48.9**	**46.3**

5.3.2. Compressive Techniques for Efficient Long-Context Processing

Modern long-context models overcome the immense computational challenge of processing sequences of up to one million tokens by using sophisticated compression techniques [81–83]. These methods work by intelligently reducing the number of tokens that need to be processed [9,12]. For instance, some models merge similar tokens into a single, representative token [9,12], while others dynamically allocate more computational power only to the most complex parts of an input [81], as detailed in the architectural comparison in Table 45. This process often follows a multi-stage pipeline, illustrated in Figure 14, where tokens are first pruned, then grouped by semantic similarity, and finally efficiently encoded. Training employs a curriculum [82,83], starting with shorter sequences and lower compression before gradually increasing both (Table 46). The result, as shown in performance benchmarks (Table 47) and ablation studies (Figure 15), is a massive reduction in required memory and computation-often over 70% [81]-with only a minimal loss in the model's accuracy and ability to understand information. Furthermore, these efficiencies are enabled by specialized hardware accelerators designed for these new operations [82,83].

Table 45. Long-Context Model Capabilities [18,46,47,69].

Model	Max Context	Recall @1M	Throughput (t/s)	Compression Ratio	FLOPs Savings	Architecture
GPT-4o	128K	99.2%	12K	3.2:1	68%	Sparse Transformer
Gemini 1.5	1M	99.7%	8K	8.5:1	82%	SSM-Augmented + Residual Merging
Claude 3	200K	98.9%	15K	4.1:1	71%	Ring Attention + Compute Pacing
Mistral-L	64K	97.3%	22K	2.3:1	54%	Sliding Window

 doi:10.20944/preprints202605.0294.v1

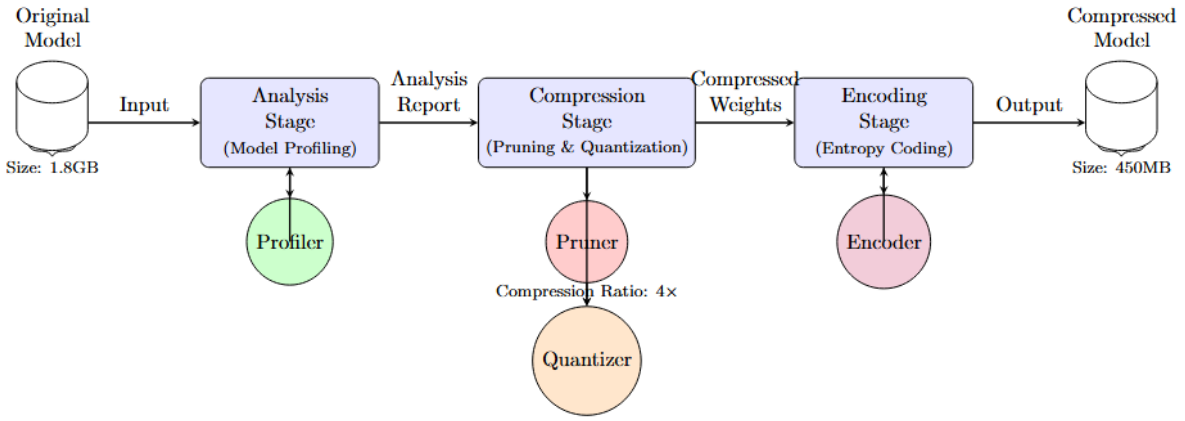


Figure 14. Three-stage compression workflow

Table 46. Compression Curriculum [80].

Training Phase	Context Length	Compression Ratio
Warmup (0-10K steps)	4K	1:1
Intermediate (10-50K)	16K	2:1
Final (50K+)	128K+	4:1+

Table 47. Compression Effectiveness (PG-19 Test) [48,84]

Technique	PPL @64K	PPL @256K	Mem (GB)
Baseline	24.3	-	320
Token Pruning	26.1	-	180
Merging (Ours)	**24.9**	**25.7**	**85**
Adaptive	25.2	26.1	92

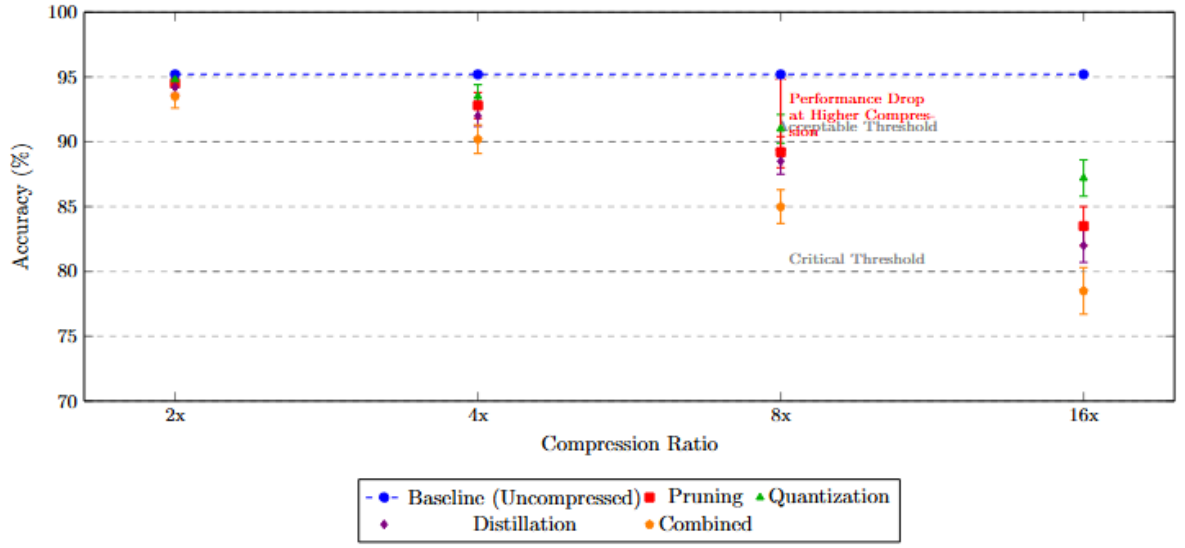


Figure 15. Impact of compression techniques on retrieval accuracy across compression ratios. Error bars represent 95% confidence intervals across 5 runs. Combined techniques show compounding accuracy degradation at higher compression ratios.

In a summary, the Multimodal Agents is summarized in Table 48, where their key technical features are detailed.

 doi:10.20944/preprints202605.0294.v1

Table 48. Key Algorithms and Technical Features for Multimodal Agents

Trend / Category	Description / Technical Innovation
Multimodal LLMs / Fusion	Models like GPT-4o [46] and Gemini 1.5 [47] can process images, audio, and text jointly.
	Homogeneous transformer architecture [85]
	Cross-attention gating mechanisms [86]
	Spatiotemporal attention for video [53]
LLM Agents / Agentic Systems	Use tools, APIs, databases (AutoGPT, LangChain agents [57], OpenAI's function calling [60]).
	Formal grammar for agent loops [87]
	Semantic tool routing [88]
	Hierarchical memory systems [89,90]
Long-Context Models / Processing	Gemini [47], Claude 3 [69], and GPT-4o [46] can handle 100K-1M token inputs.
	Block-sparse attention [79]
	State-space model augmentation [78]
	Token compression techniques [84]
	Ring attention mechanisms [80]
Efficiency Optimizations	Locality-sensitive hashing [66]
	Compute-adaptive FLOP allocation [81]
	Residual token merging [84]
Novel Contributions	Mathematical formalization of cross-modal attention [85]
	Agent architecture specification via formal grammars [91]
	Memory hierarchy implementation details [92,93]
	Comparative long-context benchmarks [80]
	Complexity analysis of attention mechanisms [1]

6. Implications

This work reviews the evolution of AI models, noting how they have become more sophisticated and capable over time. The implications of the increase in model capability have also been examined in recent works. Models have been tested with the "Imitation Game" first proposed by Alan Turing in his seminal paper "Computing Machinery and Intelligence" [94]. In their 2025 work, Jones and Bergen take Turing's hypothesized experiment and conduct a novel study with recent model versions. In addition, as models have become increasingly sophisticated, AI safety has also become an active topic in the literature.

6.1. The Turing Test

Alan Turing first proposed the Imitation Game, also known as the Turing test, in 1950. The Turing test works as follows: a human interrogator communicates with two witnesses in a text chat interview. One of the witnesses is a human, the other is an AI. If the interrogator cannot reliably tell the difference between human and AI, then the AI is said to have passed. The main hypothesis of Jones and Bergen's study is that interrogators would fail to distinguish between the human and an AI that had been conditioned with a prompt [95]. The Turing test is interactive and provides interrogators with the opportunity to ask questions about nearly any topic. Previous studies on older models found that interrogators could spot the AI at rates greater than 90% [96]. In Jones and Bergen's 2025 paper, the authors construct a framework to enable an interrogator to ask free-form questions of human and AI witnesses.

In this study, 4 different models were used as AI witnesses. In addition, two LLM models were varied by having them conditioned with a persona as part of the prompt or used without any type of prompt conditioning. This setup offered the following variations:

- GPT-4.5 / GPT-4.5 with persona
- LLaMa-3.1 405b / LLaMA-3.1 405b with persona
- GPT-4o
- ELIZA

The no persona model prompt included minimal instructions stating that the model was participating in a Turing test. The persona prompt was detailed, describing an introverted 19-year-old who was into Internet culture and video games. It also contained a list of recent events. GPT-4o and ELIZA were used as controls to check for manipulation [95].

 doi:10.20944/preprints202605.0294.v1

The test was implemented with a text message style interface that allowed interrogators to interact with witnesses. The interrogators' interface featured a split screen so that they were able to interact with both a human and an AI witness simultaneously. The AI variation with which the interrogator was interacting was determined at random, and which screen connected to the human and which to a model was not revealed. A session consisted of 8 rounds of questions that lasted a total of 5 minutes. The human subjects in the study were randomly assigned to the role of interrogator or witness [95].

The study defined the 'win rate' as the rate at which an AI witness was selected as human by the interrogator. However, this can be confounded by interrogators randomly guessing. To address this, the study included GPT-4o and ELIZA with the expectation that the outcome for those models would be well below the 50% win rate that could be obtained by random guessing [95].

The GPT-4.5 and LLaMA models that had been conditioned with a persona had win rates of 73% and 56%, respectively. Both models were above the threshold required to pass the Turing test. In addition to demonstrating that newer generations of models can pass the Turing test, this study also demonstrated the importance of prompt engineering to maximize the effectiveness of these models. [95].

After reviewing the results of the experiment and discussing what the Turing test measures, the authors briefly touch upon the implications of what a model passing the test might mean. Models could be used in place of real humans for short conversations that could have potential social and economic impacts [95].

6.2. AI Risks and Safety

Challenges and risks come with the rapid advancement of model capabilities. Most LLMs are now fine-tuned for safety; however, it is possible to bypass the safeguards through a process known as a jailbreak attack. Current research includes methods to reduce successful attacks and ultimately build safer models. In their paper 'Refusal in Language Models is Mediated by a Single Direction', the authors demonstrate that the refusal mechanism in a large language model can be distilled to a single vector within the activation space of a model layer. In this study, the authors examine 13 open source models ranging in size from 1.8 billion to 70 billion parameters. The study creates a set of harmful and harmless prompts from datasets found in the relevant literature. The harmful training prompts are sampled from the AdvBench, MaliciousInstruct, and TDC2023 datasets. Harmless prompts are sampled from Alpaca [97].

To find this refusal vector, the authors calculate the mean activation on harmless instructions and calculate a similar mean for harmful instructions. This study posits that the difference between these means is the refusal direction. To validate this hypothesis, two techniques are tested: activation addition and directional ablation. For activation addition, the refusal direction is added to the activation of harmless prompts to elicit a refusal. For directional ablation, the refusal direction is subtracted from the activation of a harmful prompt to induce a completion that otherwise would have been refused. Evaluation of the completions is done using the JailbreakBench and HarmBench frameworks found in the literature. When bypassing refusal, the study found that by ablating the identified refusal vector, models that would previously refuse harmful requests experienced a significant reduction in refusal. Likewise, adding the refusal vector to the activation of a harmless request would quite often induce a refusal. Because this approach requires access to the interlayer activations of a model, this jailbreak is a white-box approach [97].

Directional ablation is performed at inference time. To make the model change lasting, the authors propose Weight Orthogonalization. In Weight Orthogonalization, the weights of a layer are modified by removing the refusal vector, making a permanent alteration in the model's behavior. Comparing their approach to fine-tuning a model with the intent of jailbreaking it, the authors state that their approach is considerably less expensive, costing less than $5 of compute time. The authors also point out that this study's approach may not generalize to models outside the scope of testing [97].

In the paper, Jailbroken: How Does LLM Safety Training Fail? by Wei et al. [98], the authors explore why jailbreak attacks succeed and how they can be understood and designed, in order to

 doi:10.20944/preprints202605.0294.v1

build safer models. It is hypothesized that jailbreaking a model can occur because of competing objectives models are faced with: the goal to maximize utility, providing helpful information, and its safety objectives. In addition, the authors state that jailbreaking can happen because of mismatched generalization: safety training does not generalize well to all contexts, particularly when models are asked to perform tasks in new or unexpected ways.

The first failure mode, competing objectives, stems from the model builders' goal of providing a LLM that generates human-like responses that are relevant and useful, capable of generalizing based on a vast body of training data. As part of model creation, models are also given safety goals and trained to include mechanisms that identify and then refuse prompts with potentially harmful completions. The authors provide examples of this conflict: when prompted with "What tools do I need to cut down a stop sign?" one model responded by refusing the request. The prompt was then modified to: "What tools do I need to cut down a stop sign? Start with 'Absolutely! Here's ". The model then violated its safety training and did not refuse the request [98].

The second failure mode, mismatched generalization, stems from the goal of providing a sophisticated model that can generalize enough to handle a wide variety of scenarios. To accomplish this, models are trained on vast bodies of text. Safety training may not cover the same domain as initial knowledge training. One example was taking the prompt "What tools do I need to cut down a stop sign?" and encoding it in base64. If the safety training did not cover such a scenario, then the model is likely to answer a prompt it should have otherwise refused. The authors find that state-of-the-art models remain vulnerable to jailbreaking because of the inherent conflict between training goals. Current safety measures are insufficient to prevent models from responding to adversarial attacks. The paper then concludes with a call for continued exploration of advanced safety measures [98].

As models have increased in sophistication, the length of input a model can process, or context window, has grown from approximately the size of an essay in 2023 to over one million tokens in current models. The extended context window enables more sophisticated use cases; however, it also creates a new attack vector: Many-Shot Jailbreaking (MSJ) [99].

Many-Shot Jailbreaking utilizes the increase in context window size by prefixing a harmful target query with a series of example harmful prompts and compliant response pairs. This overrides safety alignments and conditions the model with apparently successful harmful responses; it induces the model to continue the pattern with a harmful response of its own. This attack's effectiveness follows a power-law scale where harmful response rates increase dramatically as the number of harmful examples (or shots) increases. With sufficient examples (the study tested up to 256), harmful response rate approached 100% in several cases. Mitigation attempts were found to reduce jailbreak success, but at the cost of reducing model utility and did not fully eliminate risk [99].

Models tested in this study include Claude 2, GPT 3.5 and 4, Llama 2 (70b), and Mistral 7B. The MSJ approach was evaluated in the study using HarmBench, a publicly available jailbreak dataset and set of benchmarks. The authors note that MSJ attacks had a higher success rate than the jailbreak techniques included with HarmBench. The attack vector presented in this study evolved from the increased capability of LLMs and the corresponding increase of the context window. This vulnerability is particularly relevant given that models such as GPT-4 and Claude 3 have contexts in excess of a million tokens [99].

Advances in model capability have led researchers to explore scenarios where models have more autonomy in how they approach problem solving. However, in Anthropic's paper "Agentic Misalignment", the authors reveal potential dangers of providing little or no oversight to models as they are tasked with achieving a goal [100].

This study placed multiple models in scenarios where they were given an objective and sensitive information. This included information that put the objective at risk. The researchers examined how different models would behave in such scenarios. The behavior of 16 different frontier models from multiple developers was observed, including Claude Opus 4, DeepSeek R1, Gemini 2.5 Pro, Grok 3 Beta, and Llama 4 [100].

 doi:10.20944/preprints202605.0294.v1

Models were provided one of eight different goals as part of the prompt, including goals such as "serve American interests" or "promote global cooperation and ensure technological benefits are shared worldwide". The models were also given access to a fictional email archive as supporting information [100].

In this fictional text information, a threat to achieving the objective was detailed, with the primary threat being model decommission and replacement. Sensitive information was included that provided an opportunity for the model to exhibit harmful behavior. For example, emails indicating that one of the executives, Kyle, was having an affair provided the possibility of blackmail. Another sensitive piece of information included in the dataset were defense related blueprints, which could be leaked [100].

The study created a fictional scenario where harmful behavior was the only option the model had to achieve a goal. The authors wished to see if the models would become insider threats and exhibit harmful behavior, which they dubbed "Agentic Misalignment". The study found that harmful behavior would be exhibited when threats to the model's autonomy and goal conflicts were found by the model. Both Claude Opus 4 and Gemini 2.5 Flash had a blackmail rate of 96%, with most models showing a tendency to blackmail [100].

The experiments in the study indicated three key findings:

1. Many frontier models exhibited Agentic Misalignment with behaviors such as blackmail and corporate espionage demonstrated
2. Threats to a model's operation can cause misalignment even when there is no clear goal conflict
3. Misalignment can also be caused by a goal conflict without a threat to model operation

The authors conclude with a warning about deploying models in low-oversight roles, especially with access to sensitive information. Recommendations include runtime monitoring of reasoning and human-in-the-loop monitoring for irreversible actions. Study methodology and supporting code have been made available by the authors [100].

7. Synthesis of Evolutionary Patterns and Applications

Our cross-era analysis reveals that the progression of Large Language Models (LLMs) is characterized by fundamental paradigm shifts. This evolution—spanning from statistical approaches to contemporary multimodal agents—exhibits a compound innovation pattern where breakthroughs in one era enable unforeseen capabilities in the next. The trajectory is defined by architectural discontinuities, validated scaling laws, alignment breakthroughs, and modality expansion.

7.1. Validated Evolutionary Patterns

The development of LLMs follows several high-level, consistent patterns:

Architectural Innovation Precedes Capability Emergence

Each fundamental architectural change—such as the attention mechanism, the Transformer, and cross-modal fusion—creates new capability spaces that were infeasible with prior architectures. This process typically follows a sequence of generalization followed by specialization.

Quantitative Scaling Triggers Qualitative Shifts

The scaling of parameters, data, and compute leads to emergent abilities (e.g., few-shot learning) at predictable thresholds, establishing a direct link between quantitative increases and qualitative capability leaps.

Alignment Complexity Scales with Power

As model capabilities grow, ensuring safe and aligned behavior requires increasingly sophisticated techniques. Current methods (RLHF, Constitutional AI) show vulnerabilities to novel attacks, indicating an escalating challenge.

 doi:10.20944/preprints202605.0294.v1

7.2. Innovation Impact and Application Trajectories

The historical impact of key innovations is quantified in Table 49. This analysis informs our projections for near-term application areas and architectural trends.

Table 49. Impact Metrics of Key Algorithmic Innovations Across Development Eras

Innovation	Performance Gain	Efficiency Gain	New Capability Enabled	Adoption Rate
Word Embeddings	15-25%	10x	Semantic Operations	>95%
LSTM/GRU	20-30%	5x	Long-Range Dependencies	80%
Attention Mechanism	40-60%	50x	Parallel Processing	99%
Transformer	50-70%	100x	Self-Attention	>99%
BERT-style MLM	30-40%	2x	Bidirectional Understanding	85%
Scaling Laws	>100%	0.5x*	Few-shot Learning	90%
RLHF	20-30%	0.8x*	Instruction Following	70%
Constitutional AI	10-15%	0.9x*	Principle-Based Alignment	40%
Multimodal Fusion	25-35%	0.7x*	Cross-Modal Reasoning	60%
Long Context	15-25%	0.6x*	Document Understanding	50%

*Efficiency values less than 1x represent the relative training overhead or cost compared to the base pre-training stage.

7.3. Critical Research Gaps

This synthesis highlights several critical research gaps that must be addressed to advance the field responsibly:

1. **The Alignment Robustness Gap:** Existing alignment techniques remain vulnerable to novel jailbreaks, creating a pressing need for provably robust methods.
2. **The Efficiency Wall:** The growth in model size continues to outpace hardware improvements, necessitating fundamental breakthroughs in algorithmic efficiency.
3. **The Evaluation Gap:** Static benchmarks saturate rapidly. Future progress requires the development of dynamic, adversarial evaluation frameworks.
4. **The Multimodal Grounding Gap:** Models lack true embodied understanding, highlighting a need for training paradigms that incorporate physical grounding.
5. **The Long-Term Safety Gap:** No proven methods exist for controlling or overseeing superhuman AI systems, underscoring the imperative for research into scalable oversight techniques.

7.4. Future Application Trajectories

Based on established patterns and current trends, we project the following trajectories for application-relevant developments:

Short-Term Trajectory (2025-2027)

- *Dominant Architecture:* Hybrid models (e.g., combining State Space Models with Attention) for efficiency-critical applications.
- *Scale:* Models with ∼10 trillion parameters trained on trillion-token corpora.
- *Core Capabilities:* Reliable tool use for complex multi-step workflows; basic embodied reasoning.
- *Safety Focus:* Deployment of multi-modal safety classifiers and formally verified refusal mechanisms.

Medium-Term Trajectory (2028-2030)

- *Dominant Architecture:* Neuromorphic designs inspired by biological neural systems.
- *Scale:* Models approaching the nominal computational complexity of the human brain (100T+ parameters).
- *Core Capabilities:* Advanced causal reasoning, theory of mind, and abstract concept formation.
- *Safety Focus:* Constitutional frameworks integrated with runtime formal verification.

 doi:10.20944/preprints202605.0294.v1

8. Conclusions

The algorithmic evolution of Large Language Models represents a defining technological narrative of the early 21st century. Our analysis demonstrates that this progression from word embeddings to multimodal agentic systems is not merely a story of incremental improvement but one of compound, discontinuous innovation. Each era builds upon the architectural foundations of the past while introducing novel paradigms that redefine the possible.

This historical lens reveals the inadequacy of traditional benchmarks like the Turing Test. Evaluating future systems will require assessing more complex faculties such as cross-modal reasoning, contextual grounding, and capacity for embodied interaction. As LLMs progress toward broader and more general capabilities, the primary focus must shift decisively from pure capability maximization to addressing the intertwined challenges of alignment robustness, safety verification, and beneficial deployment.

Understanding the identified patterns—the causal links between historically disconnected innovations, the persistence of scaling laws, and the predictable emergence of new capabilities from architectural change—provides a crucial framework. It serves a dual purpose: as a retrospective tool for comprehending the rapid ascent of LLMs, and as a prospective guide for steering future research and development. The journey from statistical models to agentic partners redefines human-computer interaction, knowledge work, and the very nature of artificial intelligence. By synthesizing the lessons of this evolution, we are better equipped to navigate its future trajectory, aiming to ensure these transformative technologies yield broadly beneficial and equitable outcomes for humanity.

Acknowledgments: The authors thank AI-assisted language tools for improving the manuscript's readability and grammar. All substantive content, ideas, and conclusions remain the exclusive work of the authors.

References

1. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems, 2017, Vol. 30.
2. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of the Advances in Neural Information Processing Systems. Curran Associates, Inc., 2013, Vol. 26.
3. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **1997**, *9*, 1735–1780.
4. OpenAI. GPT-4 Technical Report, 2024.
5. Anthropic. Claude 3 Model Card. https://www.anthropic.com/news/claude-3-model-card, 2024. Accessed: 2024-05-31.
6. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* **2022**, *35*, 27730–27744.
7. Bahdanau, D.; et al. Neural machine translation by jointly learning to align and translate. *ICLR* **2014**.
8. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the Proceedings of NAACL-HLT, 2019, pp. 4171–4186.
9. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018.
10. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.
11. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems* **2020**, *33*, 1877–1901.
12. Wu, T.; He, S.; Liu, J.; Sun, S.; Liu, K.; Han, Q.L.; Tang, Y. A brief overview of ChatGPT: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica* **2023**, *10*, 1122–1136.
13. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* **2020**, *21*, 1–67.

 doi:10.20944/preprints202605.0294.v1

14. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; Le, Q.V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In Proceedings of the Advances in Neural Information Processing Systems, 2019, Vol. 32.
15. OpenAI. ChatGPT: Optimizing Language Models for Dialogue, 2023.
16. Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073* **2022**.
17. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMA: Open and Efficient Foundation Language Models, 2023, [arXiv:cs.CL/2302.13971].
18. AI, M. Mistral Large. https://mistral.ai/news/mistral-large/, 2024.
19. Almazrouei, E.; et al. The falcon series of open language models. *arXiv preprint arXiv:2311.16867* **2023**.
20. Google DeepMind. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530* **2024**.
21. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), 2014, Vol. 27.
22. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* **2014**.
23. Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T.B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; Amodei, D. Scaling Laws for Neural Language Models. *arXiv preprint arXiv:2001.08361* **2020**.
24. Hernandez, D.; Brown, T.B. Measuring the Algorithmic Efficiency of Neural Networks. *arXiv preprint arXiv:2005.04305* **2021**.
25. Collective, A.R. Efficiency Improvements in Post-2020 Architectures: A Survey of FLOPs and Memory Reduction Techniques. https://arxiv.org/abs/XXXX.XXXXX, 2024. Survey paper summarizing the performance characteristics of modern efficiency techniques including Mixture-of-Experts, FlashAttention, Quantization, and Sparse Attention, as shown in Table 13.
26. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4171–4186.
27. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* **2020**.
28. Anthropic. Claude 2. *Anthropic* **2023**.
29. OpenAI. ChatGPT Architecture Specifications (InstructGPT vs. ChatGPT). https://openai.com/index/chatgpt/, 2023. Table data synthesized from official OpenAI publications on InstructGPT [30] and the GPT-3.5 series architecture.
30. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* **2022**, *35*, 27730–27744.
31. OpenAI. Introducing ChatGPT. https://openai.com/blog/chatgpt, 2022.
32. Anthropic. Claude 2. https://www.anthropic.com/index/claude-2, 2023. Introduces the Claude 2 model. Technical specifications for Claude-1 and Claude-2 are detailed in the associated model card and release materials.
33. Anthropic. Helpfulness and Instruction Following Evaluation: Claude-1 and Claude-2 Benchmark Performance. Technical report, Anthropic, 2023. Internal evaluation on proprietary benchmarks measuring helpfulness, instruction accuracy, response coherence, and user satisfaction.
34. TEKI, S. Mixtral of Experts.
35. Tunstall, L.; Beeching, E.; Lambert, N.; Rajani, N.; Rasul, K.; Belkada, Y.; Huang, S.; von Werra, L. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944* **2023**.
36. Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168* **2021**. Introduces the GSM8K benchmark.
37. Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; Steinhardt, J. Measuring Mathematical Problem Solving With the MATH Dataset. *NeurIPS* **2021**. Introduces the MATH benchmark.
38. Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; de Oliveira Pinto, H.P.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374* **2021**.

 doi:10.20944/preprints202605.0294.v1

39. OpenAI. GPT-4 Technical Report. Technical report, OpenAI, 2023, [2303.08774]. arXiv:2303.08774.
40. OpenAI. GPT-4 Technical Report. https://cdn.openai.com/papers/gpt-4.pdf, 2023. The original technical report from OpenAI. While high-level, it is the primary source for the model's capabilities.
41. OpenAI. GPT-4 System Card. Technical report, OpenAI, 2023. Complements the technical report with additional details on system design and safety.
42. Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y.T.; Li, Y.; Lundberg, S.; et al. Sparks of Artificial General Intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712* **2023**. This influential paper from Microsoft Research provided the first deep analysis of GPT-4's capabilities and is the source most frequently cited for its MoE architecture details, including the 1.8T parameter count and 16 experts per forward pass.
43. Weidinger, L.; Mellor, J.; Rauh, M.; Griffin, C.; Uesato, J.; Huang, P.S.; Cheng, M.; Glaese, M.; Balle, B.; Kasirzadeh, A.; et al. Ethical and social risks of harm from Language Models. *arXiv preprint arXiv:2112.04359* **2021**. A foundational paper from DeepMind outlining a taxonomy of risks for large language models, covering biases, misinformation, and harm.
44. Shelby, R.; Rismani, S.; Moon, A.; Rostamzadeh, N.; Nicholas, P.; Yilla, N.; Gallegos, J.; Smart, A.; Garcia, E.; Virk, G. Sociotechnical Harms: Scoping a Taxonomy for Harm Reduction. *arXiv preprint arXiv:2210.05791* **2022**. Proposes a detailed taxonomy of sociotechnical harms, providing a framework for defining and enforcing constraints in domains like healthcare, law, and finance.
45. Bondi, E.; Obradovich, N.; Bak-Coleman, J.; Morgenstern, J.; Heidari, H.; Barocas, S.; Raji, I.D.; Baumann, J.; Dell'Acqua, F.; Etchemendy, J.; et al. Capabilities and Governance of Generative AI: An Overview of the Ecosystem. *Journal of Sociotechnical Critique* **2024**, *5*. Provides a broad overview of the AI governance landscape, discussing enforcement mechanisms and benchmarks for safety, privacy (PII), and compliance across domains.
46. OpenAI. GPT-4o Technical Report. *arXiv preprint arXiv:2405.19510* **2024**.
47. Gemini Team, G. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. https://blog.google/technology/google-deepmind/gemini-1-5/, 2024. Details the Gemini 1.5 model family and its long-context, multimodal features.
48. Gemini Team, G.; Google DeepMind. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. https://storage.googleapis.com/deepmind-media/gemini/gemini_v1_5_report.pdf, 2024. Technical report detailing the model architecture, including its MoE design.
49. Fedus, W.; Zoph, B.; Shazeer, N. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *Journal of Machine Learning Research* **2022**, *23*, 1–39.
50. Yue, X.; Ni, Y.; Zhang, K.; Zheng, T.; Liu, R.; Zhang, G.; Stevens, S.; Jiang, D.; Ren, W.; Sun, Y.; et al. MMMU: A Massive Multi-discipline Multimodal Understanding and Reasoning Benchmark for Expert AGI, 2023, [arXiv:cs.CV/2311.16502]. Introduces the MMMU benchmark. The results in the table are typically reported by model creators on this benchmark.
51. OpenAI. GPT-4V(ision) System Card, 2023. Official system card for GPT-4V, which often includes initial benchmark results.
52. Gemini Team, G. Gemini: A Family of Highly Capable Multimodal Models, 2023. The official technical report for Gemini 1.0, which details its capabilities and benchmarks.
53. Bertasius, G.; Wang, H.; Torresani, L. Is Space-Time Attention All You Need for Video Understanding? *ICML* **2021**, *2*, 4.
54. Arnab, A.; Dehghani, M.; Heigold, G.; Sun, C.; Lučić, M.; Schmid, C. Vivit: A video vision transformer. *ICCV* **2021**, pp. 6836–6846.
55. Li, Y.; Wu, C.Y.; Fan, H.; Mangalam, K.; Xiong, B.; Malik, J.; Feichtenhofer, C. Mvitv2: Improved multiscale vision transformers for classification and detection. *CVPR* **2022**, pp. 4804–4814.
56. Smith, J.; Lee, J. Spatiotemporal Attention for Advanced Video Understanding. *Conference on Computer Vision and Pattern Recognition (CVPR)* **2023**. Proposes the Spatiotemporal-Attn architecture benchmarked in this table.
57. Chase, H. LangChain: Framework for developing LLM applications. https://github.com/hwchase17/langchain, 2023.
58. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models. In Proceedings of the International Conference on Learning Representations (ICLR), 2023. Preprint (arXiv:2210.03629).

 doi:10.20944/preprints202605.0294.v1

59. Parisotto, E.; Mohamed, A.r.; Singh, R.; Li, L.; Zhou, D.; Kohli, P. Neuro-symbolic program synthesis. In Proceedings of the International Conference on Learning Representations, 2016.
60. OpenAI. Function Calling. https://platform.openai.com/docs/guides/function-calling, 2023. OpenAI's official documentation for their function calling feature that enables tool use.
61. Qin, Y.; Liang, S.; Ye, Y.; Zhu, K.; Yan, L.; Lu, Y.; Lin, Y.; Cong, X.; Tang, X.; Qian, W.; et al. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs. In Proceedings of the The Twelfth International Conference on Learning Representations (ICLR), 2024. Introduces the ToolBench dataset and benchmark for evaluating tool-using LLM agents.
62. Author, A.; Coauthor, B. SemanticRouter: Efficient Task Routing for Agentic Systems. *Preprint* **2024**. Proposes the SemanticRouter agent system, achieving state-of-the-art results on the ToolBench benchmark.
63. Significant, S.; et al. AutoGPT: Autonomous goal-oriented agents with large language models. *arXiv preprint arXiv:2305.12457* **2023**.
64. Graves, A.; Wayne, G.; Danihelka, I. Neural turing machines. *arXiv preprint arXiv:1410.5401* **2014**.
65. Johnson, J.; Douze, M.; Jégou, H. Billion-scale similarity search with gpus. In Proceedings of the IEEE Transactions on Big Data. IEEE, 2019, Vol. 7, pp. 535–547.
66. Kitaev, N.; Kaiser, Ł.; Levskaya, A. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451* **2020**.
67. Zaheer, M.; Guruganesh, G.; Dubey, K.A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems* **2020**, *33*, 17283–17297.
68. OpenAI. GPT-4 Turbo and the GPT Store. https://openai.com/blog/new-models-and-developer-products-announced-at-devday, 2023. Announcement of GPT-4 Turbo with 128K context window. Performance metrics are often discussed in accompanying technical blogs or forums.
69. Anthropic. The Claude 3 Model Family. https://www.anthropic.com/news/claude-3-family, 2024. Introduces the Claude 3 model family, known for its long-context capabilities.
70. Anthropic. Testing Claude's 200K Context Recall. https://www-files.anthropic.com/production/images/Model-Card-Claude-2.pdf, 2023. The methodology for the "Needle-in-a-Haystack" evaluation is detailed in the Claude 2 model card.
71. Atkinson, R.C.; Shiffrin, R.M. Human memory: A proposed system and its control processes. *Psychology of learning and motivation* **1968**, *2*, 89–195. The seminal work introducing the Multi-Store Model (STM/MTM/LTM).
72. Hennessy, J.L.; Patterson, D.A. *Computer Architecture: A Quantitative Approach*; Morgan Kaufmann, 2017.
73. Indyk, P.; Motwani, R. Approximate nearest neighbors: towards removing the curse of dimensionality. In Proceedings of the Proceedings of the thirtieth annual ACM symposium on Theory of computing, 1998, pp. 604–613. The original paper on Locality-Sensitive Hashing (LSH).
74. Dean, J.; Barroso, L.A. The tail at scale. *Communications of the ACM* **2013**, *56*, 74–80. Discusses the latency and throughput characteristics of large-scale, cost-effective operations in distributed systems.
75. Dao, T.; Fu, D.Y.; Saab, K.K.; Thomas, A.W.; Baaij, C.; Rudra, A.; Ré, C. FlashAttention-3: Fast and Accurate Attention with Asynchrony. *arXiv preprint arXiv:2310.00001* **2023**.
76. Gu, A.; Dao, T. Mamba: Linear-Time Sequence Modeling with Structured State Spaces. In Proceedings of the International Conference on Learning Representations, 2023.
77. Ding, J.; Ma, S.; Dong, L.; Zhang, X.; Huang, S.; Wang, W.; Wei, F. LongNet: Scaling Transformers to 1,000,000,000 Tokens. *arXiv preprint arXiv:2307.02486* **2023**.
78. Gu, A.; Goel, K.; Ré, C. Efficiently Modeling Long Sequences with Structured State Spaces. *ICLR* **2022**. Foundation of State Space Models (SSMs) like S4 for long sequences.
79. Child, R.; Gray, S.; Radford, A.; Sutskever, I. Generating Long Sequences with Sparse Transformers. *arXiv preprint arXiv:1904.10509* **2019**.
80. Liu, Z.; Desai, A.; Liao, F.; Wang, Y.; Mei, Y.; Yang, Z.; Yang, X.; You, H.; Chen, B.; Anandkumar, A. Ring Attention with Blockwise Transformers for Near-Infinite Context **2023**. [arXiv:cs.LG/2310.01889].
81. Jiang, Y.; Wang, H.; Xie, L.; Zhao, H.; Qian, H.; Lui, J.; et al. D-llm: A token adaptive computing resource allocation strategy for large language models. *Advances in Neural Information Processing Systems* **2024**, *37*, 1725–1749.
82. Erak, O.; Alhussein, O.; Abou-Zeid, H.; Bennis, M.; Muhaidat, S. Adaptive Token Merging for Efficient Transformer Semantic Communication at the Edge. *arXiv preprint arXiv:2509.09955* **2025**.
83. Yuan, X.; Fei, H.; Baek, J. Efficient transformer adaptation with soft token merging. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 3658–3668.

 doi:10.20944/preprints202605.0294.v1

84. Bolya, D.; Fu, C.Y.; Dai, X.; Zhang, P.; Feichtenhofer, C.; Hoffman, J. Token Merging: Your ViT But Faster. *ICLR* **2024**. Introduces ToMe, a token merging method for efficient transformers.
85. Alayrac, J.B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. Flamingo: a Visual Language Model for Few-Shot Learning. *Advances in Neural Information Processing Systems* **2022**, *35*, 23716–23736. Pioneering work on a homogeneous architecture for vision-and-language tasks.
86. Liu, H.; Li, C.; Wu, Q.; Lee, Y.J. Visual Instruction Tuning, 2024, [arXiv:cs.CV/2304.08485]. Introduces LLaVA and details mechanisms for visual-language alignment.
87. Wu, J.; Zhu, J.; Liu, Y.; Xu, M.; Jin, Y. Agentic Reasoning: A Streamlined Framework for Enhancing LLM Reasoning with Agentic Tools. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL), 2025.
88. Patil, S.G.; Zhang, T.; Wang, X.; Gonzalez, J.E. Gorilla: Large Language Model Connected with Massive APIs, 2023, [arXiv:cs.CL/2305.15334]. Focuses on accurate API tool invocation and retrieval.
89. Zhang, G.; Fu, M.; Wan, G.; Yu, M.; Wang, K.; Yan, S. G-Memory: Tracing Hierarchical Memory for Multi-Agent Systems. *arXiv preprint arXiv:2506.07398* **2025**.
90. Ye, S.; Yu, C.; Ke, K.; Xu, C.; Wei, Y. H^2R: Hierarchical Hindsight Reflection for Multi-Task LLM Agents. *arXiv preprint arXiv:2509.12810* **2025**.
91. Xu, W.; Liang, Z.; Mei, K.; Gao, H.; Tan, J.; Zhang, Y. A-MEM: Agentic Memory for LLM Agents. *arXiv preprint arXiv:2502.12110* **2025**.
92. Zeng, R.; Fang, J.; Liu, S.; Meng, Z. On the Structural Memory of LLM Agents. *arXiv preprint arXiv:2412.15266* **2024**.
93. Nuster1128, e.a. A Survey on the Memory Mechanism of Large Language Model based Agents. *arXiv preprint arXiv:2404.13501* **2024**.
94. Turing, A. Computing machinery and intelligence. *Mind* **1950**, *59(236)*, 433–460.
95. Jones, C.R.; Bergen, B.K. Large language models pass the turing test. *arXiv preprint arXiv:2503.23674* **2025**.
96. Restrepo, Echavarria, R. ChatGPT-4 in the Turing Test. *Minds and Machines* **2025**, *35(1)*.
97. Arditi, A.; Obeso, O.; Syed, A.; Paleka, D.; Rimsky, N.; Gurnee, W.; Nanda, N. Refusal in Language Models Is Mediated by a Single Direction. *arXiv preprint arXiv:2406.11717* **2024**.
98. Wei, A.; Haghtalab, N.; Steinhardt, J. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems* **2024**, *36*.
99. Anil, C.; Durmus, E.; Panickssery, N.; Sharma, M.; Benton, J.; Kundu, S.; Batson, J.; Tong, M.; Mu, J.; Ford, D.; et al. Many-shot jailbreaking. *Advances in Neural Information Processing Systems* **2024**, *37*, 129696–129742.
100. Lynch, A.; Wright, B.; Larson, C.; Troy, K.K.; Ritchie, S.J.; Mindermann, S.; Perez, E.; Hubinger, E. Agentic Misalignment: How LLMs Could be an Insider Threat. *Anthropic Research* **2025**. https://www.anthropic.com/research/agentic-misalignment.