Review

Not peer-reviewed version

Large Language Models: A Survey of Architectures, Training Paradigms, and Alignment Methods

Deepshikha Bhati *, Fnu Neha, Devi Sri Bandaru, Matthew Weber, Ishan Dilipbhai Gajera

Posted Date: 15 January 2026

doi: 10.20944/preprints202601.1138.v1

Keywords: Large Language Models (LLMs); generative AI; transformer-based models; multimodal learning; foundation models

 doi:10.20944/preprints202601.1138.v1

Review

Large Language Models: A Survey of Architectures, Training Paradigms, and Alignment Methods

Deepshikha Bhati *, Fnu Neha, Devi Sri Bandaru, Matthew Weber and Ishan Dilipbhai Gajera

Department of Computer Science, Kent State University, Kent, OH 44242
* Correspondence: dhati@kent.edu

Abstract

Large Language Models (LLMs) have become foundational to modern Artificial Intelligence (AI), enabling advanced reasoning, multimodal understanding, and scalable human-AI interaction across diverse domains. This survey provides a comprehensive review of major proprietary and open-source LLM families, including GPT, LLaMA 2, Gemini, Claude, DeepSeek, Falcon, and Qwen. It systematically examines architectural advancements such as transformer refinements, mixture-of-experts paradigms, attention optimization, long-context modeling, and multimodal integration. The paper further analyzes alignment and safety mechanisms, encompassing instruction tuning, reinforcement learning from human feedback, and constitutional frameworks, and discusses their implications for controllability, reliability, and responsible deployment. Comparative analysis of training strategies, data curation practices, efficiency optimizations, and application settings highlights key trade-offs among scalability, performance, interpretability, and ethical considerations. Beyond synthesis, the survey introduces a structured taxonomy and a feature-driven comparative study of over 50 reconstructed LLM architectures, complemented by an interactive visualization interface and an open-source implementation to support transparency and reproducibility. Finally, it outlines open challenges and future research directions related to transparency, computational cost, data governance, and societal impact, offering a unified reference for researchers and practitioners developing large-scale AI systems.

Keywords: Large Language Models (LLMs); generative AI; transformer-based models; multimodal learning; foundation models

1. Introduction

Large Language Models (LLMs) have become central to contemporary artificial intelligence (AI), enabling progress in natural language processing (NLP), reasoning, and multimodal understanding. Built on transformer-based architectures and trained on large-scale heterogeneous data, LLMs show strong generalization across tasks including text generation, question answering, code synthesis, summarization, and multimodal inference. These capabilities have driven adoption across domains such as scientific research, education, healthcare, creative workflows, and software engineering.

As LLMs evolve from experimental systems to widely deployed infrastructure, architectural and training design choices increasingly affect computational efficiency, reliability, safety, and regulatory compliance. Consequently, systematic and comparative analysis of LLM design decisions has become essential not only for advancing model development but also for enabling responsible and sustainable real-world deployment.

1.1. Historical Background: From GPT-2 to Modern Multimodal LLMs

Modern LLM emerged with the release of GPT-2 in 2019, which showed that large-scale unsupervised pretraining could yield coherent and context-aware language generation [1]. This was followed

 doi:10.20944/preprints202601.1138.v1

by GPT-3 (175 billion (B) parameters), whose increased scale revealed emergent few-shot and zero-shot capabilities, reshaping evaluation practices and downstream adaptation strategies [2].

Subsequent developments have shifted emphasis from parameter scaling alone toward architectural refinement and efficiency. Open and semi-open models such as LLaMA 2 emphasized accessible training strategies [3], while Gemini and Claude introduced advances in long-context modeling, multimodal reasoning, and alignment [4,5]. In parallel, models including DeepSeek, Qwen, and Falcon explored mixture-of-experts routing, optimized attention mechanisms, and instruction-following behavior under computational constraints [6–9].

A defining recent trend is the transition toward multimodal architectures. Models such as GPT-4o and Gemini 1.5 integrate text, vision, audio, and video within unified frameworks using cross-modal attention and expert specialization [4,10]. This evolution reflects a broader shift from text-centric language modeling toward general-purpose systems capable of grounded and cross-domain reasoning. Figure 1 highlights major milestones in LLM development from 2019 to 2025.

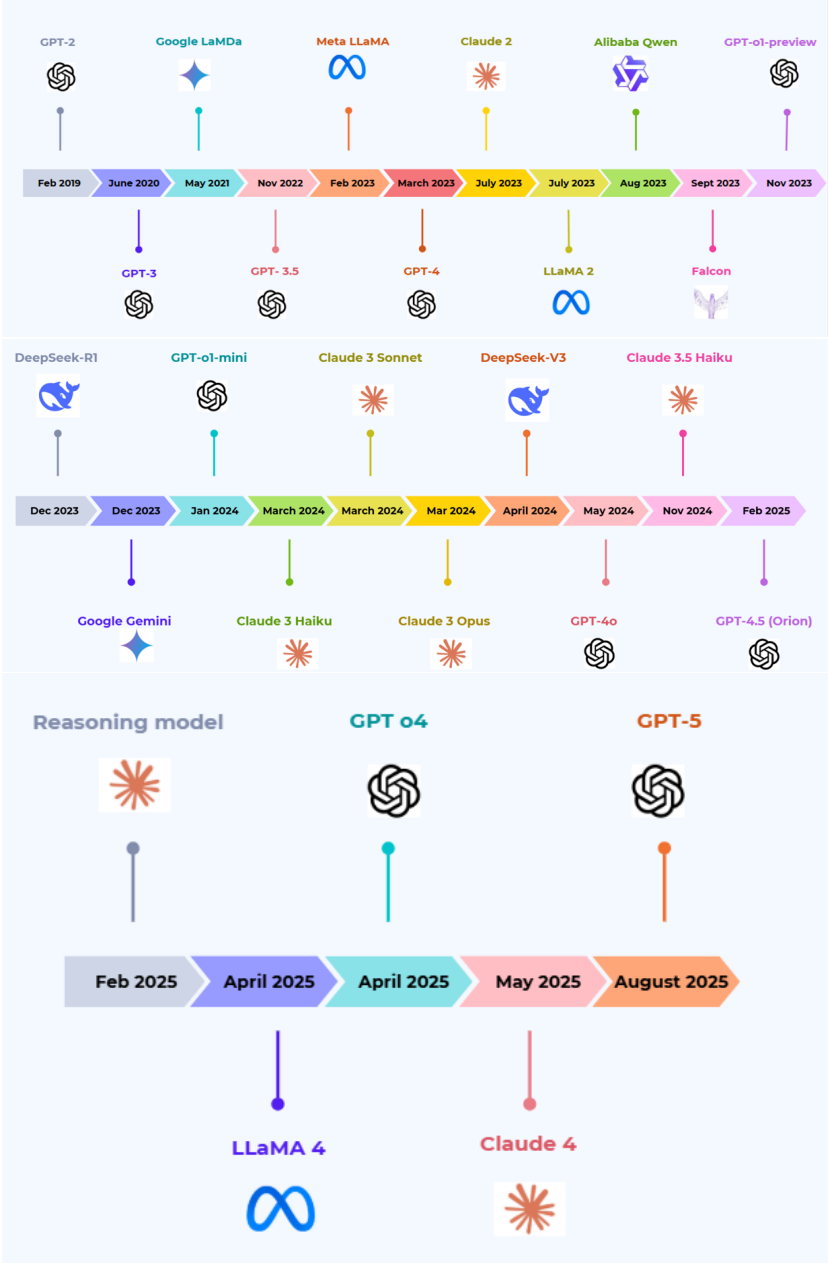

Figure 1. Timeline of major Large Language Model releases (2019–2025).

 doi:10.20944/preprints202601.1138.v1

1.2. Goals of This Paper

This survey presents a comparative analysis of seven widely adopted LLM families: OpenAI's GPT series, Meta's LLaMA 2, Google's Gemini, Anthropic's Claude, DeepSeek, Qwen AI by Alibaba, and Falcon by the Technology Innovation Institute. Models are examined through a unified analytical framework covering architectural design, training pipelines, alignment strategies, multimodal capabilities, and reported performance characteristics. In addition to synthesizing prior work, this study provides reconstructed architectural representations and an interactive analysis framework to support transparent and reproducible comparison across model families.

The main contributions of this work are summarized as follows:

- A systematic consolidation of existing literature into a coherent taxonomy of contemporary LLM architectures spanning proprietary and open-source ecosystems.
- Reconstruction and comparative analysis of over 50 representative LLM architectures, abstracted from technical reports and model documentation to enable consistent cross-model comparison.
- An interactive, feature-driven visualization interface that operationalizes the proposed taxonomy, enabling selective inspection and side-by-side comparison across architectural, training, capability, and safety dimensions.
- Release of an open-source implementation and live web-based deployment of the LLM Model Explorer to support transparency, reproducibility, and continued community-driven exploration.
- Comparative analysis of core design choices, including attention mechanisms, normalization strategies, activation functions, context-length scaling, and efficiency-oriented optimizations.
- Examination of alignment and safety methodologies, including Reinforcement Learning from Human Feedback (RLHF) and constitutional approaches, with emphasis on instruction adherence and hallucination mitigation.
- Identification of open challenges related to transparency, computational cost, data governance, and responsible deployment.

By organizing current LLM research around shared design dimensions rather than model-specific reported findings, this survey aims to provide a concise reference for understanding the current LLM landscape and to inform future work on scalable and trustworthy AI systems.

2. Related Work

Research on LLMs builds on prior advances in neural architectures, large-scale pretraining, and human-aligned learning. This section situates the present survey within existing literature on transformer-based models, generative AI surveys, alignment methodologies, and multimodal learning.

2.1. Foundational Transformer Models and LLMs

Transformer architecture introduced by Vaswani et al. [11] established self-attention (SA) as a scalable mechanism for modeling long-range dependencies and enabled efficient parallel training. This architectural paradigm supported early autoregressive language models such as GPT and GPT-2 [1], showing the effectiveness of large-scale unsupervised pretraining.

Subsequent models, most notably GPT-3 [2], showed emergent zero-shot and few-shot learning capabilities, motivating further exploration into scaling behavior, optimization strategies, and data diversity. Later iterations, including GPT-3.5 and GPT-4 [12], incorporated alignment objectives and architectural refinements to improve instruction adherence and robustness.

In parallel, open and semi-open model families such as LLaMA [3], Gemini [4], Claude [13], DeepSeek [14], and Falcon [9] expanded the LLM design space through innovations in attention optimization, positional encoding, mixture-of-experts (MoEs) routing, and multimodal integration. These efforts highlight distinct trade-offs among efficiency, scalability, alignment, and deployment constraints.

 doi:10.20944/preprints202601.1138.v1

2.2. Surveys on Generative AI, Alignment, and Multimodal Learning

Several surveys have assessed the broader landscape of generative AI and foundation models. Bommasani et al. [15] framed foundation models as a unifying paradigm with wide-ranging societal implications, emphasizing challenges related to interpretability, fairness, and governance.

Alignment-oriented research has been reviewed through studies on RLHF [16] and Constitutional AI [13], which address controllability, safety, and value alignment in large-scale models. The emergence of multimodal LLMs has been documented through work on models such as Flamingo [17] and technical reports on GPT-4 [12], highlighting mechanisms for cross-modal attention, vision–language fusion, and expert routing.

Although these surveys provide important perspectives on individual dimensions of LLM development, they majorly emphasize specific paradigms, such as alignment, scaling, or multimodality, rather than offering a unified architectural comparison across multiple model families.

2.3. Positioning of This Work

This survey presents a comparative analysis of major large language model (LLM) families across proprietary and open-source ecosystems, examining architectural design choices, training pipelines, alignment strategies, multimodal capabilities, and reported performance within a single analytical framework. The analysis highlights shared trends, divergent design philosophies, and open challenges that inform future research directions.

The survey focuses exclusively on core LLM architectures and training paradigms. System-level extensions—including retrieval-augmented generation, tool-augmented agents, and multi-agent systems—are excluded, as they extend beyond the base language model and would confound direct architectural comparison.

3. Taxonomy of Large Language Model (LLM) Architectures

The rapid evolution of Large Language Models (LLMs) has given rise to a diverse and expanding design space encompassing architectural innovations, training methodologies, and alignment mechanisms. Rather than converging toward a single canonical architecture, contemporary LLMs explore a spectrum of design trade-offs aimed at balancing scalability, computational efficiency, long-context reasoning, multimodal integration, and safety. In this section, we introduce a structured taxonomy that organizes modern LLMs according to their architectural evolution, core building blocks, and optimization strategies, providing a unified framework for systematic comparison across major model families.

To ground this taxonomy in concrete design choices, we reconstructed architectural diagrams for the LLM families analyzed in this survey using publicly available technical reports, model cards, and peer-reviewed research publications. These diagrams are intentionally abstracted and model-agnostic, emphasizing high-level architectural components, such as attention mechanisms, normalization strategies, expert routing, context-handling techniques, and multimodal extensions, while omitting implementation-specific details. This abstraction enables direct and consistent comparison of architectural decisions across both proprietary and open-source models.

To further support transparency, reproducibility, and interactive exploration, we developed an online visualization interface that operationalizes the proposed taxonomy through feature-driven model inspection. For any selected LLM, the interface provides a *Select Features* panel that allows users to explicitly select or deselect individual categories of information, including *Developer*, *Architecture*, *Training Data*, *Parameters*, *Attention Mechanism*, *Text Generation*, *Language Translation*, *Code Generation*, *Image Generation*, *Video Generation*, *Summarization*, *Question Answering*, *Safety Features*, *Creativity*, *Versatility*, *Privacy Concerns*, *Bias and Misinformation*, and the *Architecture Diagram*. A *Select All* option is provided to enable full model inspection by default.

Figure 2 illustrates the feature-selectable inspection view of the LLM Model Explorer, demonstrating how users can dynamically enable or disable specific architectural and functional attributes

 doi:10.20944/preprints202601.1138.v1

for a selected model. Beyond surveying individual models, a central contribution of this work is the systematic reconstruction and comparative analysis of over 50 large language model architectures spanning proprietary and open-source ecosystems. We operationalize the proposed taxonomy through a publicly accessible interactive visualization interface, which is provided alongside this study to support transparent, feature-driven exploration of model architectures, training strategies, and capabilities. To ensure reproducibility and extensibility, we also release the complete implementation of the LLM Model Explorer as open-source code, together with a live web deployment, enabling continued community use and future model integration. The source code is available at https://github.com/Devisri-B/LLM-Architectures, and the interactive interface can be accessed at https://devisri-b.github.io/LLM-Architectures/.

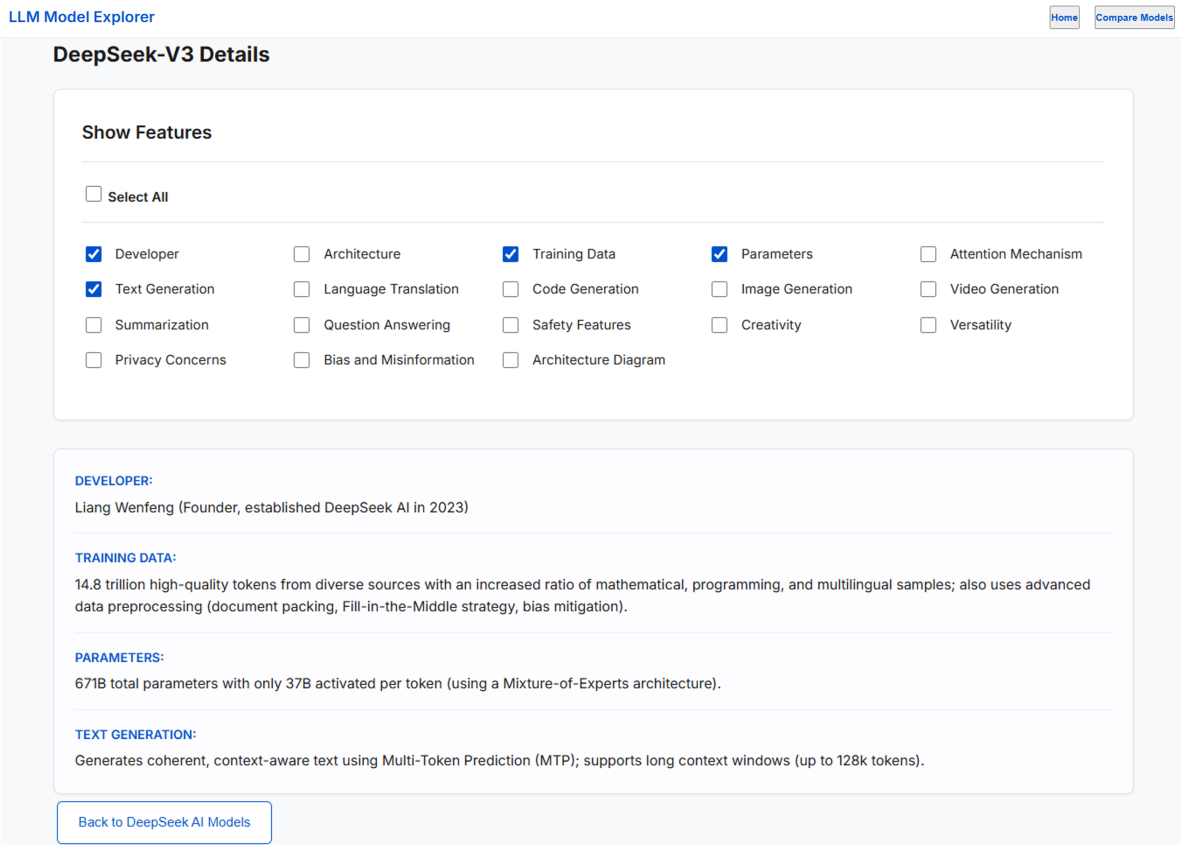

Figure 2. Feature-selectable model inspection view in the LLM Model Explorer. Users can dynamically select or deselect architectural and functional attributes for a chosen LLM (shown for DeepSeek-V3) to support targeted exploration and comparison.

In addition to single-model inspection, the LLM Model Explorer supports feature-based comparison between multiple models. Users can select two LLMs and choose a subset of features of interest, after which the interface presents a side-by-side comparison of the selected attributes in a structured table. This comparison view enables focused analysis of architectural and functional differences between models while maintaining consistency across feature categories. Figure 3 illustrates the feature-based comparison view for two selected LLMs.

Based on the selected features, the interface dynamically updates the displayed content, showing only the corresponding textual descriptions and architectural visualizations. This selective disclosure mechanism allows users to focus on specific aspects of a model while minimizing cognitive overload, thereby supporting targeted analysis of individual LLMs as well as consistent, side-by-side comparison across proprietary and open-source model families. By coupling a formal architectural taxonomy with an interactive, feature-selectable visualization interface, this work extends beyond a static survey of LLM architectures. Instead, it provides a reusable analytical and exploratory resource that enables

 doi:10.20944/preprints202601.1138.v1

systematic understanding, comparison, and reasoning about the rapidly evolving design landscape of large language models.

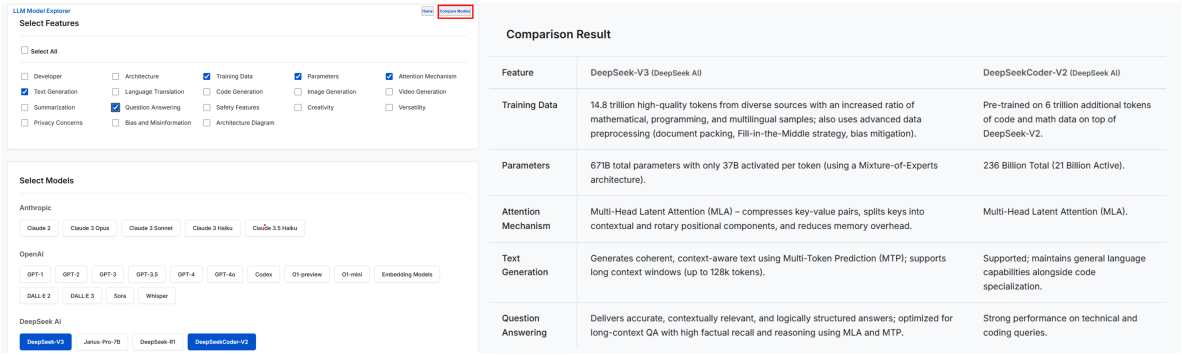

Figure 3. Interactive feature-based side-by-side comparison of two large language models in the LLM Model Explorer, enabled through the *Compare Models* interface, allowing users to analyze differences across training data, parameters, attention mechanisms, and task capabilities.

3.1. Transformer Evolution and Architectural Enhancements

Nearly all contemporary LLMs are built upon the Transformer architecture introduced by Vaswani et al. [11]. By replacing recurrence with SA, Transformers enable parallel computation and effective modeling of long-range dependencies. Early decoder-only models, such as GPT-2 [1], showed the suitability of this architecture for autoregressive language generation. Subsequent work has focused on addressing efficiency and scalability constraints through targeted architectural refinements.

Representative extensions include:

- **Grouped Query Attention (GQA)**, adopted in LLaMA 2 [3], which reduces inference cost by sharing key–value projections across groups of attention heads.
- **MoE** routing, employed in Gemini [4], which enables sparse activation of expert subnetworks and parameter-efficient scaling.
- **Multiquery and Multigroup Attention**, used in Falcon [18], which lowers key–value cache memory requirements during inference.
- **Multi-Head Latent Attention (MLA)**, introduced in DeepSeek [14], which applies low-rank compression to attention representations to reduce computation and storage overhead.

These enhancements reflect a gradual refinement of the Transformer architecture to support larger models, longer context windows, and multimodal inputs under practical resource constraints.

3.2. Core Architectural Components

3.2.1. Positional Encodings

Transformers require explicit positional information to model token order. Early models such as GPT-2 used fixed sinusoidal encodings. Most recent LLMs, including LLaMA 2, Qwen AI, and Falcon, adopt Rotary Positional Embeddings (RoPE) [19], which encode relative position directly within the attention mechanism and generalize more effectively to long sequences. Gemini and DeepSeek extend this approach through interpolation-aware scaling, enabling substantially larger context lengths.

3.2.2. Self-Attention (SA) Variants

SA remains primary for contextual reasoning in LLMs, but several variants have been proposed to improve efficiency and expressiveness:

- **Multiquery Attention**, used in Falcon and Qwen, which shares key and value projections across heads to reduce memory usage.
- **GQA**, employed in LLaMA 2, which balances efficiency and representational capacity.
- **Cross-Modal Attention**, implemented in Gemini, which integrates textual, visual, and auditory inputs within a unified attention framework.

 doi:10.20944/preprints202601.1138.v1

These adaptations illustrate how attention mechanisms are adapted to large-scale and multimodal modeling requirements.

3.2.3. Activation Functions

Activation functions influence both representational capacity and training stability. While earlier Transformer models relied on `ReLU` or `GELU`, many modern LLMs employ gated variants. SwiGLU, adopted in LLaMA 2 and Qwen AI, improves gradient flow and empirical stability in deep architectures.

3.2.4. Normalization Strategies

Normalization is important for stable optimization in deep Transformers. Earlier architectures commonly used Post-LayerNorm, whereas recent LLMs, including LLaMA 2 and Claude, favor RMSNorm with pre-normalization [20]. RMSNorm reduces computational overhead by omitting mean subtraction and has been shown to improve convergence behavior at scale.

3.3. Optimization and Alignment Techniques

Training and deploying LLMs at scale relies on a combination of optimization and alignment strategies:

- **AdamW optimization**, typically combined with cosine learning-rate decay and linear warm-up schedules [21].
- **Mixed-precision training**, using formats such as `bfloat16` or FP8 to improve training efficiency while maintaining numerical stability.
- **Gradient clipping and Z-loss regularization**, applied in large-scale models such as Falcon-180B to stabilize optimization under large batch sizes [18].
- **Instruction tuning and RLHF**, employed by models including GPT-4, Claude, and DeepSeek to align outputs with human preferences and safety constraints [16].

Together, these techniques support scalable training, robust instruction-following behavior, and improved safety properties across modern LLMs. Building on this taxonomy, the following sections analyze representative LLM families across proprietary and open-source ecosystems using a common analytical framework encompassing architectural design, training methodology, alignment strategy, and reported capabilities.

4. GPT-2 Model

GPT-2, introduced by OpenAI, represents an early large-scale autoregressive language model that showed the effectiveness of unsupervised pretraining on large text corpora for general-purpose language modeling [1]. Built on a Transformer decoder architecture, GPT-2 established several design and training principles that influenced subsequent generations of LLMs.

4.1. Model Architecture

GPT-2 adopts a decoder-only Transformer composed of stacked SA and feed-forward layers. Multi-head SA enables each token to attend to prior tokens in the sequence, supporting contextual dependency modeling over long spans. Residual connections and layer normalization are applied throughout the network to stabilize optimization and enable deeper architectures. The model supports a maximum context length of 1,024 tokens and scales to 1.5B parameters, allowing it to capture complex syntactic and semantic regularities (Figure 4).

 doi:10.20944/preprints202601.1138.v1

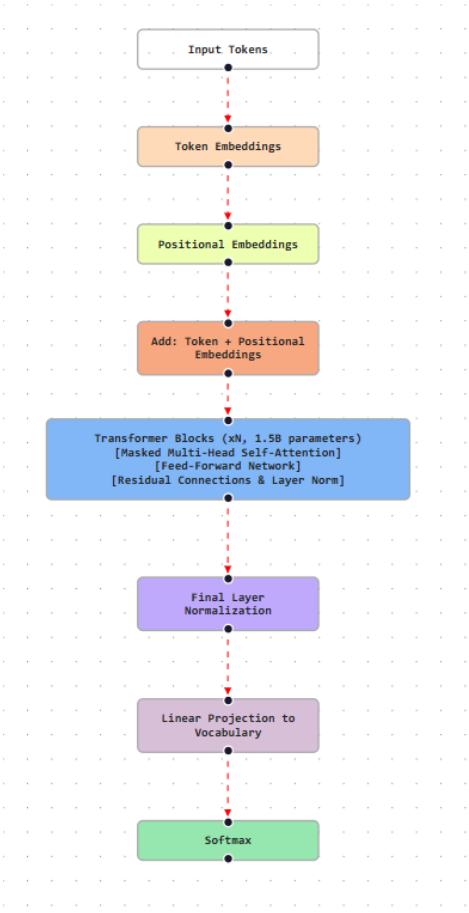

Figure 4. GPT-2 architecture based on a decoder-only Transformer.

4.2. Data and Preprocessing

GPT-2 employs byte-level Byte Pair Encoding (BPE) to provide language-agnostic tokenization and efficient vocabulary representation. By operating directly on byte sequences, this approach avoids character-level fragmentation while maintaining a compact base vocabulary of 256 tokens. The model is trained on the WebText dataset, which contains over 8 million (M) documents (approximately 40 GB) curated from outbound Reddit links. Automated filtering methods, including Dragnet, are used to remove boilerplate content and duplicates, improving data quality.

4.3. Training and Alignment

GPT-2 is pretrained using an autoregressive next-token prediction objective on the WebText corpus. Optimization is performed with the Adam optimizer and a learning rate schedule incorporating warm-up and linear decay. Training is distributed across multiple GPUs to support large-scale computation.

Although GPT-2 shows strong zero-shot and few-shot generalization, it can be further adapted through supervised fine-tuning for downstream tasks such as summarization, translation, and sentiment analysis. Fine-tuning adjusts pretrained representations to task-specific distributions while preserving general linguistic competence.

5. GPT-3 Model

GPT-3 represents a major scaling advance in large language modeling, demonstrating that increased parameterization and training data diversity yield emergent zero-shot, one-shot, and few-shot capabilities [2]. By substantially expanding model size and corpus breadth, GPT-3 set new performance benchmarks across a broad range of natural language processing tasks.

 doi:10.20944/preprints202601.1138.v1

5.1. Model Architecture

GPT-3 retains a decoder-only Transformer architecture while extending scale and depth. The model family includes eight variants ranging from 125M to 175B parameters, with the largest configuration comprising 96 Transformer layers. The maximum context length is increased to 2,048 tokens, enabling improved coherence over longer sequences. Architectural scaling and subsequent refinements are illustrated in Figure 5.

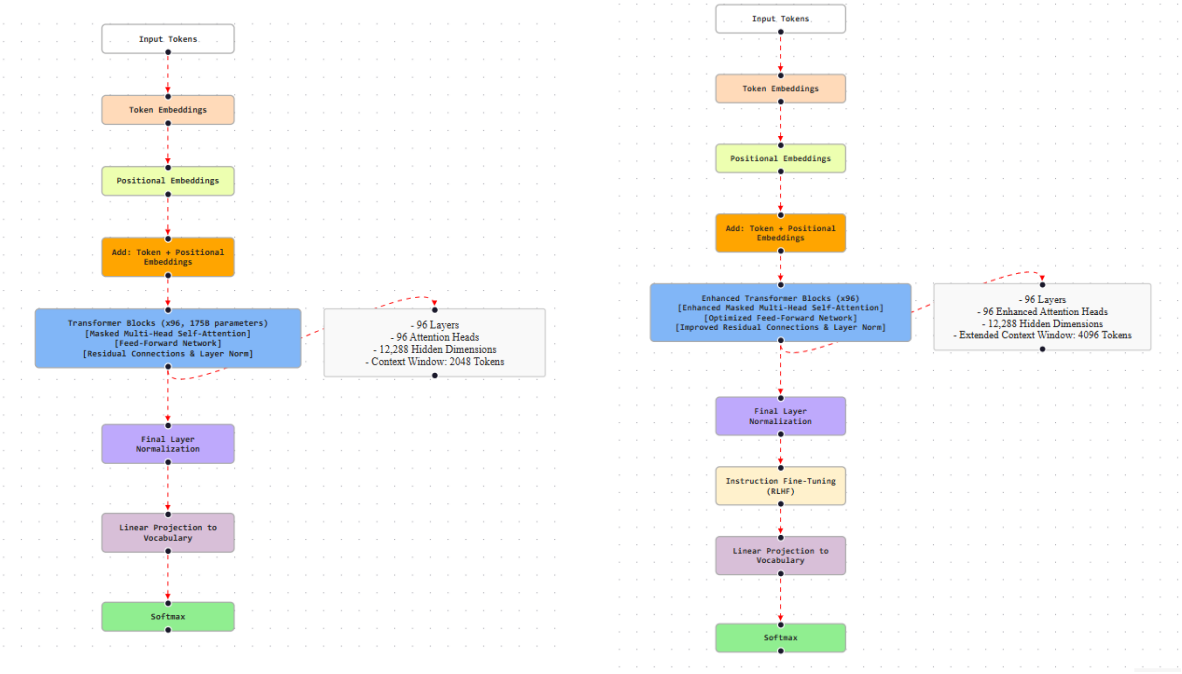

Figure 5. Architectural evolution from GPT-3 to GPT-3.5. The left panel illustrates the scaled Transformer architecture of GPT-3, while the right panel highlights refinements introduced in GPT-3.5, including instruction tuning and task-oriented adaptations.

5.2. Data and Preprocessing

GPT-3 is trained on a heterogeneous mixture of large-scale text sources, including filtered Common Crawl data, English Wikipedia, Books1, Books2, and an expanded WebText corpus. Automated filtering and fuzzy deduplication reduce redundancy and mitigate data leakage. Higher-quality datasets are upsampled during training to increase their influence. Tokenization is performed using Byte Pair Encoding, supporting efficient representation of rare terms and multilingual content.

5.3. Training and Alignment

GPT-3 is trained with an autoregressive language modeling objective on approximately 300B tokens. Optimization employs adaptive learning rates scaled to model size, with larger configurations using lower learning rates and larger batch sizes to maintain training stability. Training is distributed across large GPU clusters using a combination of data, model, and pipeline parallelism.

Although GPT-3 is primarily evaluated in few-shot settings without explicit fine-tuning, supervised fine-tuning can be applied for domain-specific adaptation. Such adaptation improves task-level performance but introduces trade-offs related to overfitting and potential erosion of general-purpose knowledge.

6. GPT-3.5 Model

GPT-3.5 represents a refinement of the GPT-3 lineage that prioritizes instruction adherence, response coherence, and improved task alignment through enhanced training procedures rather than architectural redesign [22]. The model family illustrates how alignment-oriented optimization can substantially improve usability and controllability without increasing model scale.

 doi:10.20944/preprints202601.1138.v1

6.1. Model Architecture

GPT-3.5 retains the decoder-only Transformer architecture of GPT-3 while incorporating attention-level optimizations and instruction-tuned variants. Models such as `text-davinci-002` and `text-davinci-003` are optimized for following complex natural language instructions, whereas `code-davinci-002` is specialized for programming-related tasks. These variants improve output consistency, controllability, and task execution across diverse use cases (Figure 5).

6.2. Training and Alignment

The training corpus for GPT-3.5 extends earlier datasets with more recent and diverse sources, including technical documentation and large-scale code repositories. Increased exposure to programming data enhances reasoning over syntactic and semantic code structures, while broader textual coverage improves cross-domain generalization.

Instruction tuning is a defining component of GPT-3.5 training. Instruction-tuned variants exhibit improved sensitivity to user prompts, more reliable adherence to constraints, and more structured output generation. These properties support deployment in applications such as conversational systems, educational tools, technical writing, and creative content generation.

7. GPT-4 Model

GPT-4 represents an advanced stage in the evolution of LLMs, extending prior Transformer-based systems with improved reasoning performance, broader generalization, and multimodal capability [23, 24]. Developed by OpenAI, GPT-4 builds on earlier GPT models while introducing support for joint text–image inputs and longer context handling. Although detailed architectural specifications remain proprietary, publicly disclosed information indicates several design and training enhancements that contribute to its performance across language understanding, generation, and vision–language tasks.

7.1. Model Architecture

GPT-4 retains a Transformer-based architecture centered on SA for contextual dependency modeling. A distinguishing feature is multimodal input support, enabling the model to process both text and image inputs within a unified framework. This capability facilitates tasks requiring integrated visual and linguistic reasoning. GPT-4 also supports an expanded context window relative to earlier GPT models, improving coherence over extended inputs and interactions. In addition, the model shows stronger reasoning performance on tasks involving logical inference, mathematical problem solving, and complex instruction following. These developments are illustrated in Figure 6.

 doi:10.20944/preprints202601.1138.v1

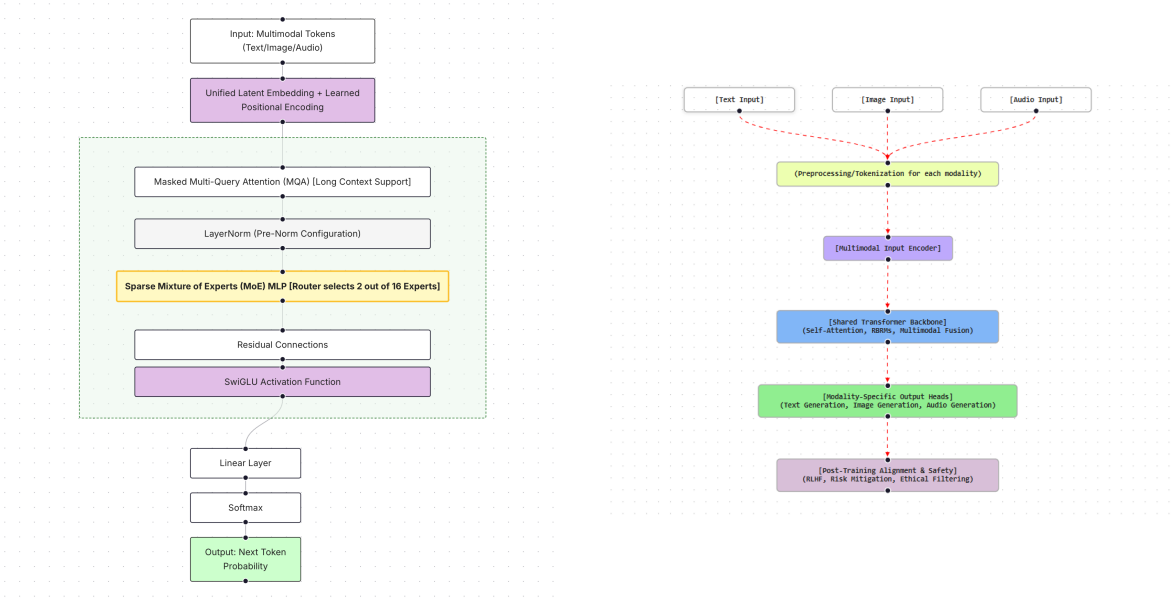

Figure 6. Architectural progression from GPT-4 to GPT-4o. The left panel depicts GPT-4 with multimodal text–image reasoning, while the right panel highlights GPT-4o's unified multimodal design supporting text, vision, audio, and video inputs.

7.2. Data and Preprocessing

The training corpus for GPT-3.5 builds upon earlier GPT-3 datasets, with additional instruction-focused data and task-specific fine-tuning. While OpenAI has not publicly disclosed detailed dataset composition, post-training procedures emphasize instruction adherence, response coherence, and controllability.

7.3. Training and Alignment

GPT-4 training follows a two-stage process comprising large-scale pretraining and post-training alignment. During pretraining, the model is optimized using an autoregressive next-token prediction objective on large unlabeled datasets, including both text-only and vision–language data. This phase supports the acquisition of linguistic, semantic, and cross-modal representations.

Post-training alignment is performed using RLHF. Human annotators generate prompts and rank model outputs based on quality, relevance, and safety, producing a reward model that captures human preferences. The base model is then optimized using Proximal Policy Optimization (PPO) to maximize the learned reward signal. Additional safety constraints are incorporated to reduce harmful or biased outputs. Iterative feedback from deployment and expert evaluation further refines model behavior, contributing to improved robustness and alignment.

8. GPT-4o Model

GPT-4o is a multimodal LLM developed by OpenAI that supports integrated processing and generation across text, code, images, audio, and video [25]. Building on the design principles of GPT-4, GPT-4o extends multimodal integration to both inputs and outputs, enabling unified reasoning across heterogeneous content types. The model is trained on large-scale multimodal datasets collected prior to late 2023 and demonstrates broad generalization across linguistic, visual, and audiovisual tasks. Its multimodal integration capabilities are illustrated in Figure 6.

8.1. Training and Alignment

GPT-4o is trained on a large and diverse corpus spanning multiple data modalities. Textual data are drawn primarily from publicly available web documents, supporting general language

 doi:10.20944/preprints202601.1138.v1

understanding, multilingual competence, and stylistic flexibility. Additional curated datasets covering programming languages and mathematical reasoning contribute to improved performance in code generation and symbolic problem solving. Multimodal training data include images, audio recordings, and video segments, incorporated in both paired and unpaired formats to support cross-modal representation learning.

The training pipeline follows a multi-stage approach. During initial self-supervised pretraining, GPT-4o is optimized on unlabeled multimodal data to learn shared representations across text, vision, and audio streams. This phase establishes foundational multimodal understanding and generation capabilities. Post-training alignment combines supervised fine-tuning with RLHF to improve accuracy, contextual appropriateness, and alignment with human preferences. Prior to deployment, the model undergoes extensive safety evaluation to identify and mitigate risks related to harmful, misleading, or unsafe content generation.

8.2. Safety and Compliance Measures

GPT-4o incorporates multiple safety and compliance mechanisms to support responsible deployment. Automated content moderation systems are used to detect and restrict harmful outputs, including hate speech, explicit material, violent content, and exposure of sensitive personal information. Privacy-preserving measures are also supported, such as opt-out mechanisms for image data, allowing users to exclude visual content from future training. These safeguards are designed to enhance reliability, user trust, and ethical use in multimodal AI applications.

9. GPT-O1 Model

The GPT-O1 model family represents a recent generation of Transformer-based LLMs developed by OpenAI, with emphasis on improved reasoning performance, alignment, and safety across diverse application settings [26]. It includes specialized variants such as `O1-preview` and `O1-mini`, which are optimized for different performance and deployment constraints. Building on established Transformer foundations, GPT-O1 introduces refinements in training, alignment, and reinforcement learning that enhance contextual understanding, structured reasoning, and safe response generation. Representative variants are shown in Figure 7.

 doi:10.20944/preprints202601.1138.v1

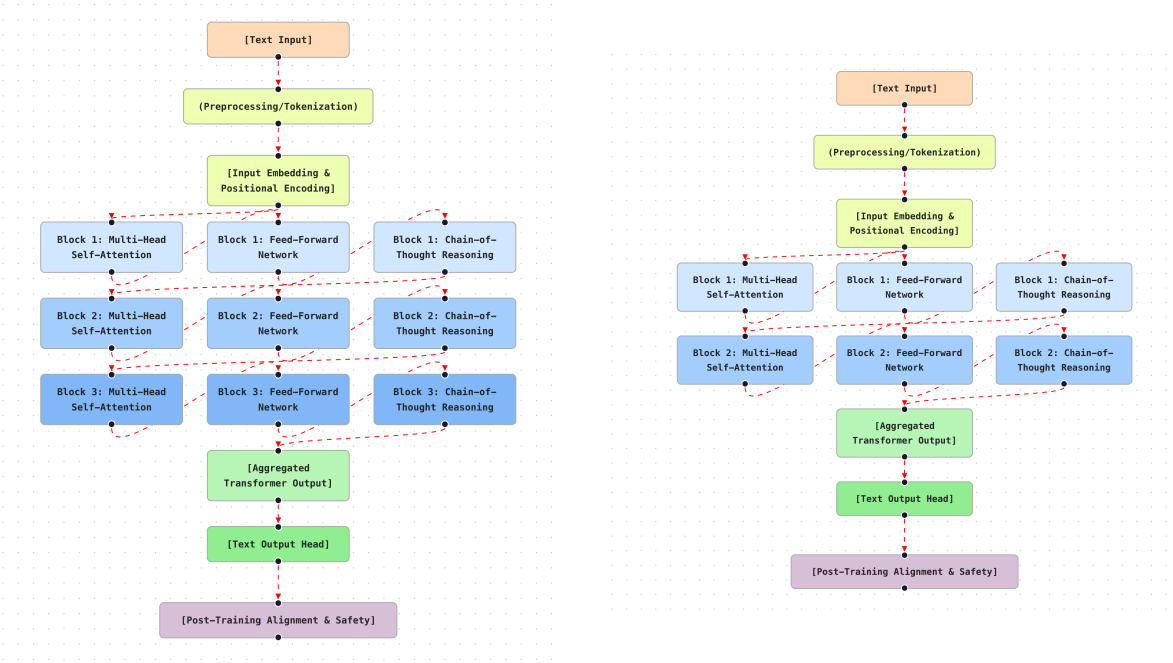

Figure 7. Architectural variants within the GPT-O1 family. The left panel illustrates GPT-O1-preview, designed for advanced reasoning and robustness, while the right panel shows GPT-O1-mini, optimized for low-latency and resource-efficient deployment.

9.1. Model Architecture

GPT-O1 retains a decoder-only Transformer architecture augmented to support complex reasoning and controlled dialogue generation. Multi-head SA enables integration of multiple contextual signals, supporting coherent responses across extended interactions. It supports chain-of-thought-style reasoning, allowing the model to internally represent intermediate reasoning steps before producing final outputs. This design improves performance on tasks requiring multi-step inference, including mathematical reasoning, code synthesis, and strategic decision-making.

9.2. Training and Alignment

GPT-O1 is trained using a multi-stage pipeline combining large-scale pretraining with reinforcement-based alignment. Pretraining leverages a heterogeneous corpus that includes web text, academic literature, reasoning benchmarks, and domain-specific sources, supplemented by licensed and proprietary data. Curated datasets emphasizing logical consistency, factual accuracy, and ethical response behavior are incorporated to strengthen reasoning depth and safety.

Post-training alignment relies heavily on RLHF, aligning model outputs with human preferences and safety objectives. Reinforcement learning mechanisms are integrated throughout training to encourage robust reasoning, adaptability, and stable behavior under diverse prompting conditions.

9.3. Data Processing and Safety

Extensive data preprocessing is applied to improve generalization and reduce risk. Filtering pipelines remove duplicate samples, low-quality content, and sensitive personal information. Safety-oriented classifiers and moderation tools are integrated during training to limit exposure to harmful or inappropriate content. These measures contribute to reducing unsafe outputs and improving compliance with deployment requirements.

9.4. Safety and Model Variants

Safety considerations are integral to GPT-O1 development. The model is trained to comply with OpenAI alignment and safety frameworks, with particular attention to robustness against adversarial

 doi:10.20944/preprints202601.1138.v1

prompting and misuse. Continuous monitoring and iterative refinement support adaptation to evolving usage patterns and ethical expectations.

The GPT-O1 family includes variants tailored to distinct deployment scenarios. `O1-preview` targets high-demand applications requiring strong reasoning capability and robustness, such as advanced analytics and software development. `O1-mini` prioritizes efficiency and low latency, providing competitive performance under constrained computational resources. Together, these variants extend the applicability of the GPT-O1 family across research and production environments.

Table 1 provides a comparative overview of key AI and multimodal models developed by OpenAI, including language models (e.g., GPT-2, GPT-3, GPT-4), image and video generators (e.g., DALL·E 2, DALL·E 3, Sora), audio models (e.g., Whisper), and specialized reasoning or code-generation systems (e.g., Codex, GPT-4o, and the O1 family). The table summarizes each model's architecture, intended purpose, supported input/output modalities, parameter scale (where publicly available), and distinguishing features, offering a compact reference for understanding design trade-offs and capabilities across modalities [27]. Figure 8 further illustrates representative architectural paradigms for two OpenAI models, Sora and Whisper, highlighting how multimodal generation and perception extend beyond text-centric architectures summarized in Table 1.

Table 1. Compact Comparison of AI and Multimodal Models.

Model	Arch.	Purpose	I/O	Params	Special Features
GPT-2	Transformer	Text generation	Text/Text	1.5B	First OpenAI transformer-based model
GPT-3	Transformer	Text generation	Text/Text	175B	Large-scale autoregressive pretraining
GPT-3.5	Transformer	Enhanced generation	Text/Text	175B	Instruction tuning and better reasoning
GPT-4	Transformer-Multimodal	Text + vision	Txt+Img/Txt+Img	1.76T	Larger context window, multimodal
DALL-E 2	Enc-Dec Transformer	Text-to-image	Text/Image	3.5B	CLIP-based latent inversion, photo-quality
DALL-E 3	Transformer	Image generation	Text/Image	–	Better captions, prompt alignment
Sora	Diffusion Transformer	Text-to-video	Txt+Img/Vid	–	3D coherence, simulation, editing
Whisper	Transformer (Audio)	Speech recog.	Audio/Text	798M	Multilingual, VAD, translation
Codex	GPT-3 variant	Code generation	Text/Code	12B	Multi-language programming
GPT-4o	Multimodal Transformer	Unified multimodal	Txt+Img+Aud/All	–	Emotion, real-time audio-vision-text
O1-preview	Transformer (CoT)	Reasoning + safety	Text/Text	–	Chain-of-thought, jailbreak resistance
O1-mini	Optimized Transf.	Fast reasoning	Text/Text	–	Low-latency, coding-focused

 doi:10.20944/preprints202601.1138.v1

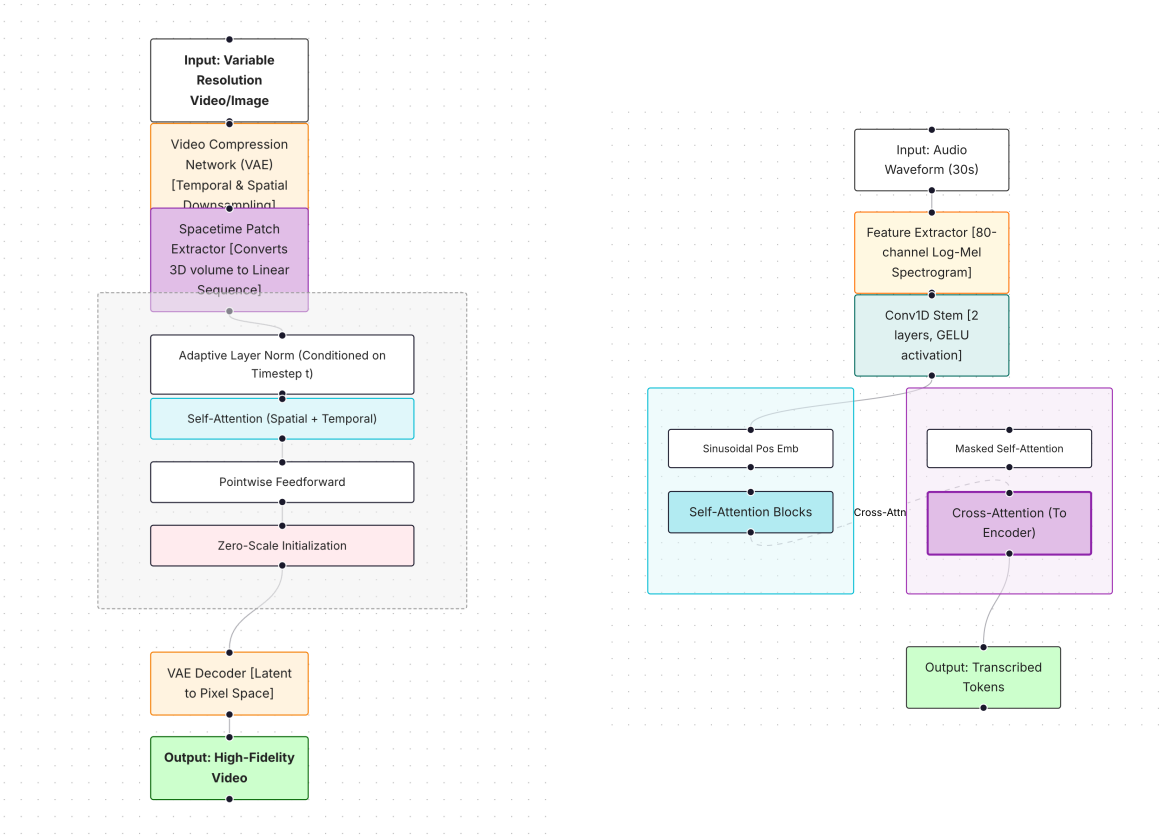

Figure 8. Representative multimodal architectures from OpenAI. **Left:** Sora video generation architecture, leveraging spatiotemporal latent representations and joint attention for high-resolution video synthesis. **Right:** Whisper speech recognition model, employing an encoder–decoder Transformer with cross-attention to transcribe audio into text.

10. LLaMA 2 Model

LLaMA 2, developed by Meta in collaboration with Microsoft, is a prominent open-access LLM designed for both research and commercial use [28]. The model emphasizes transparency and reproducibility through public disclosure of architectural choices, training procedures, and alignment methods, enabling broad adoption for downstream research and application development.

 doi:10.20944/preprints202601.1138.v1

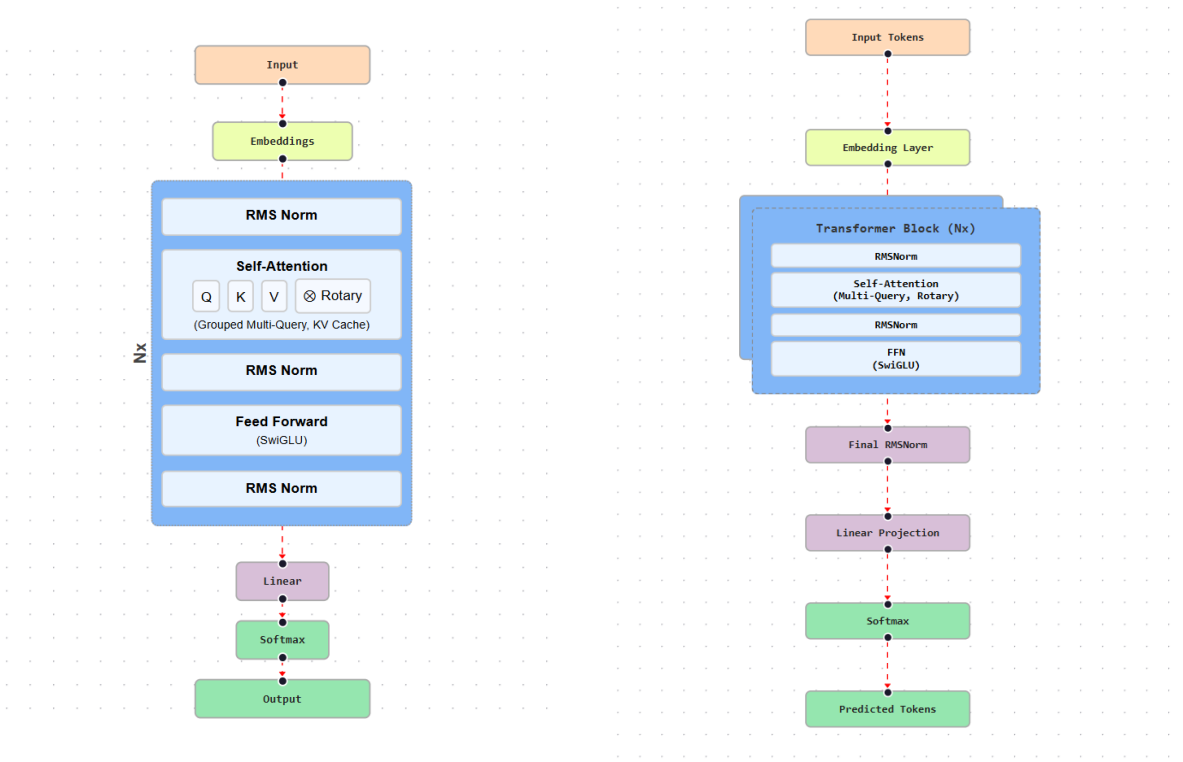

Figure 9. Architectural evolution of the LLaMA family. The left panel shows the original LLaMA architecture, while the right panel highlights enhancements introduced in LLaMA 2, including improved normalization, attention efficiency, and instruction retention mechanisms.

10.1. Model Architecture and Innovations

LLaMA 2 is based on a decoder-only Transformer architecture with refinements targeting training stability, inference efficiency, and long-context reasoning. The model adopts RMSNorm with pre-normalization to stabilize optimization in deep networks and employs SwiGLU activation functions to improve expressiveness and convergence. Rotary Positional Embeddings (RoPE) encode positional information directly within the attention mechanism, supporting improved generalization to longer sequences.

Inference efficiency is enhanced through GQA, which reduces key–value memory overhead. The maximum context length is extended to 4,096 tokens, improving performance on long-form generation and document-level reasoning tasks. Figure 9 illustrates the architectural progression from LLaMA to LLaMA 2.

10.2. Model Configurations

LLaMA 2 is released in configurations ranging from 7B to 70B parameters, enabling trade-offs between performance, memory footprint, and latency [29]. These configurations support deployment across heterogeneous hardware environments. Table 2 summarizes the correspondence between LLaMA and LLaMA 2 model sizes.

 doi:10.20944/preprints202601.1138.v1

Table 2. Model Size Comparison Between LLaMA and LLaMA 2.

LLaMA 1	**LLaMA 2**
LLaMA 7B	LLaMA 2 7B
LLaMA 13B	LLaMA 2 13B
LLaMA 33B	LLaMA 2 70B
LLaMA 65B	–

10.3. Training Data and Optimization

LLaMA 2 models are trained on approximately two trillion tokens drawn from a mixture of publicly available and licensed text sources, explicitly excluding data from Meta's proprietary platforms [3]. This constraint supports transparency and reduces privacy-related concerns.

Training employs the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-5}$, and a weight decay of 0.1, combined with learning-rate warm-up and cosine decay schedules. Computational efficiency is improved through FlashAttention-style kernels and activation checkpointing. Optimization is distributed across NVIDIA A100 GPUs using tensor parallelism.

10.4. Alignment and Instruction Retention

LLaMA 2 incorporates Ghost Attention (GAtt) to improve instruction retention across multi-turn interactions. During fine-tuning, user instructions are provided only at the start of a dialogue, while higher loss weight is assigned to subsequent turns to encourage sustained adherence. RLHF is applied using synthetic multi-turn dialogues, reinforcing responses favored by a learned reward model [3]. Figure 10 summarizes this alignment pipeline and illustrates the architectural adaptations used in LLaMA 2–Instruct and Code LLaMA.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 January 2026 doi:10.20944/preprints202601.1138.v1

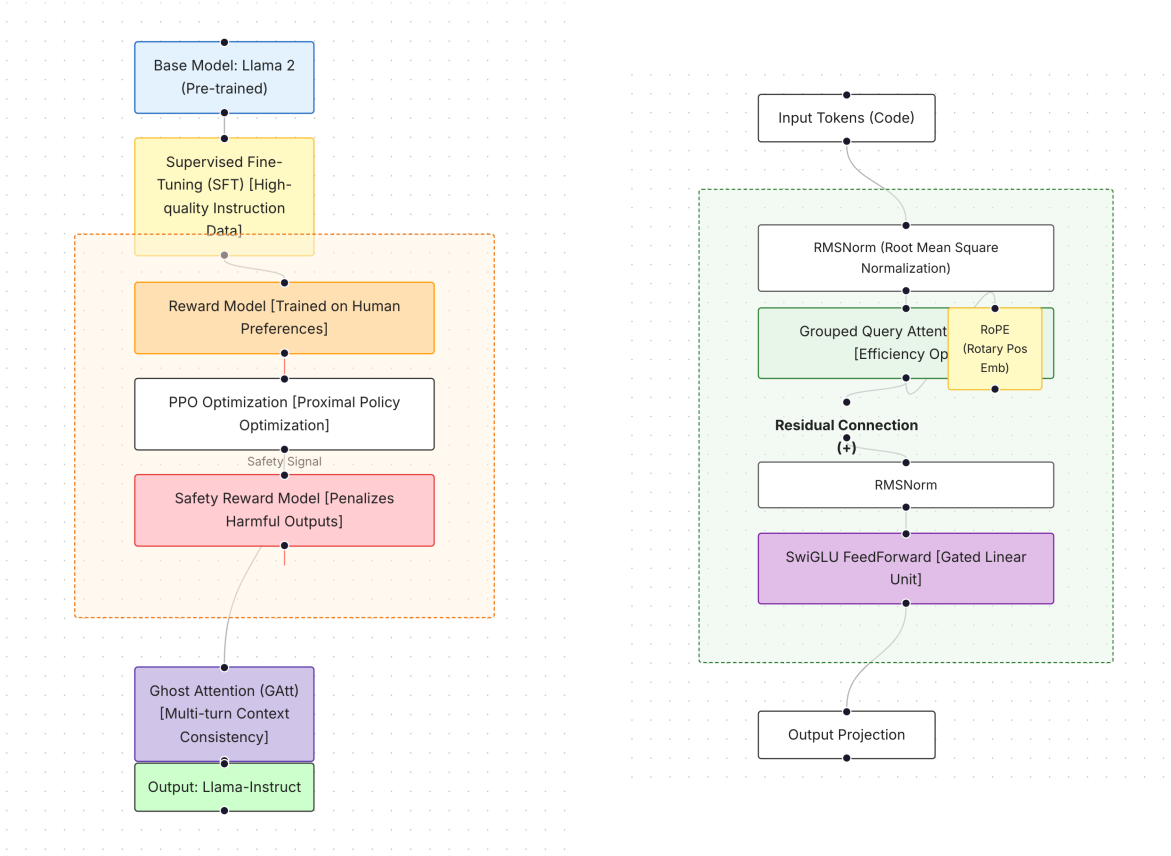

Figure 10. Task-specialized adaptations of LLaMA 2. **Left:** LLaMA 2–Instruct training pipeline incorporating supervised fine-tuning, reward modeling, RLHF, and Ghost Attention for sustained instruction adherence. **Right:** Code LLaMA architecture optimized for code understanding and generation using grouped-query attention, RoPE, and SwiGLU feedforward layers.

10.5. Applications

LLaMA 2 supports a wide range of language understanding and generation tasks, including summarization, dialogue, and long-form text generation [30]. In machine translation, LLaMA-based systems such as ALMA achieve competitive performance using limited parallel data [31]. Code LLaMA extends the family with variants optimized for code generation and infilling across multiple programming languages [32]. Additional studies report strong performance in clinical summarization and complex question answering, with ongoing work exploring multimodal extensions [33].

10.6. Safety, Bias, and Trust

Safety and ethical considerations are integral to LLaMA 2 development. Fine-tuning with RLHF aligns outputs with human preferences, while safety-oriented reward models penalize harmful or misleading content [3,34]. Evaluation against benchmarks such as TruthfulQA and ToxiGen assesses factual accuracy and robustness to unsafe generation [35].

Bias mitigation includes fairness-aware data filtering and explicit exclusion of personally identifiable information during training. Additional fine-tuning on factual corpora and alignment objectives reduces hallucination rates and improves reliability in high-stakes settings, including healthcare and education [36].

11. Google's Gemini

Gemini is a multimodal LLM developed by Google DeepMind that natively integrates text, image, audio, video, and code understanding [37]. Emerging from the integration of Google Brain and

 doi:10.20944/preprints202601.1138.v1

DeepMind, Gemini reflects a unified effort to develop scalable, versatile, and safety-aligned foundation models capable of operating across heterogeneous modalities and deployment contexts.

11.1. Model Architecture and Modular Design

Gemini is built on a Transformer-based architecture with native multimodal support. Modality-specific encoders process inputs such as text, images, and code, while cross-modal attention mechanisms fuse these representations into a shared latent space. This design enables joint reasoning across modalities for tasks including multimodal question answering, image captioning, and code synthesis [38]. The overall architecture is illustrated in Figure 11.

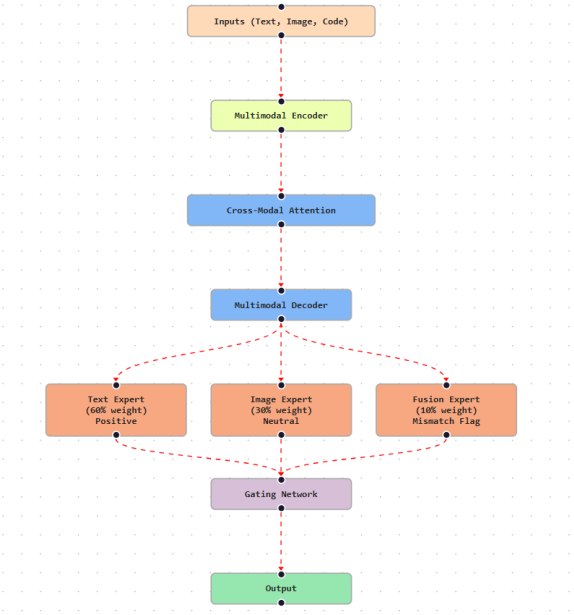

Figure 11. Architecture of Google's Gemini multimodal model.

Gemini 1.5 extends this framework through MoE design comprising specialized Text, Image, and Fusion experts coordinated by a gating network. This routing mechanism dynamically allocates computation based on input composition, enabling efficient specialization without uniformly increasing model cost [39].

11.2. Model Configurations and Training Data

Gemini is trained on a mixture of publicly available web data and proprietary interaction data, although detailed dataset composition has not been fully disclosed [40]. This limited transparency has prompted discussion regarding standardized documentation for large-scale training corpora.

Gemini family includes configurations optimized for different deployment scenarios. Lightweight variants such as Gemini Nano-1 (1.8B parameters) and Nano-2 (3.25B parameters) target on-device inference for resource-constrained platforms. Larger configurations are designed for cloud-based deployment, supporting enterprise-scale and research-oriented workloads [38].

11.3. Attention Mechanisms and Efficiency Optimizations

Gemini employs a decoder-only Transformer augmented with efficiency-oriented attention mechanisms. Multi-Query Attention reduces memory overhead by sharing key and value projections across heads, improving inference efficiency. Flash Attention and Flash Decoding further reduce memory access costs and latency during training and inference, particularly in streaming and real-time applications [41]. These optimizations are aligned with Google's TPU-based infrastructure to support large-scale deployment.

 doi:10.20944/preprints202601.1138.v1

11.4. Capabilities and Applications

Gemini demonstrates strong performance across language, code, and multimodal tasks. Gemini 1.5 supports context windows of up to 1M tokens, enabling long-range reasoning, document-level understanding, and sustained contextual coherence. The model performs competitively on benchmarks such as MMLU and Natural2Code, reporting low perplexity and strong BLEU scores [42].

In code generation, Gemini achieves a HumanEval Pass@1 score of 74.9% in the Pro variant and 74.7% on Natural2Code with Gemini-Ultra [42]. Multilingual support extends to over 100 languages, enabling low-latency translation and cross-lingual applications when deployed on Google Cloud infrastructure [43]. Beyond text-centric tasks, Gemini's multimodal reasoning supports applications in summarization, question answering, image and video generation, and educational content creation [39].

11.5. Safety, Privacy, and Trust

Safety and responsible deployment are integral to Gemini's design. The model incorporates layered safeguards, including content moderation, response filtering, and prompt-level controls, to mitigate harmful or inappropriate outputs [40]. On-device deployment of Gemini Nano variants further enhances privacy by limiting reliance on cloud-based inference and reducing data exposure [41].

Gemini has faced scrutiny related to bias and misinformation in sensitive contexts. In response, Google has strengthened alignment procedures, moderation strategies, and evaluation frameworks to improve fairness, reliability, and trustworthiness in real-world deployments.

12. Claude

Claude is a family of LLMs developed by Anthropic, with a design emphasis on safety, alignment, and interpretability. Rather than prioritizing raw performance alone, Claude models focus on controllability, ethical behavior, and adherence to user intent. The models are deployed across multiple platforms, including Quora Poe, Notion, DuckDuckGo, and Anthropic's developer API, supporting both research and production use.

12.1. Claude 2

Claude 2 is a decoder-only Transformer model optimized for long-context reasoning and document-level understanding [44]. The architecture incorporates standard components such as multi-head SA, positional embeddings, residual connections, and layer normalization. With support for context lengths of up to 100,000 tokens, Claude 2 is well suited for tasks involving extended documents and sustained multi-turn interaction. The model architecture is shown in Figure 12.

 doi:10.20944/preprints202601.1138.v1

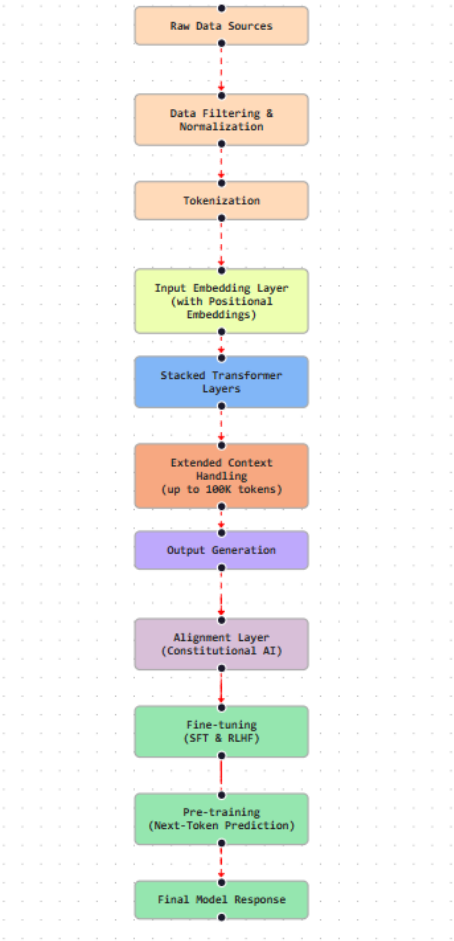

Figure 12. Claude 2 architecture based on a decoder-only Transformer.

The training pipeline includes text normalization, deduplication, and filtering of low-quality or harmful content, followed by subword tokenization using Byte Pair Encoding. Approximately 10% of the training data is non-English, enabling multilingual capability. Pretraining employs an autoregressive next-token prediction objective optimized with adaptive methods such as Adam and curriculum-based scheduling.

Post-training alignment combines supervised fine-tuning, RLHF, and Anthropic's Constitutional AI framework. Constitutional AI enables the model to revise outputs according to predefined ethical principles, improving alignment while reducing reliance on extensive human annotation [45]. Additional safeguards include red-teaming, adversarial testing, and bias mitigation.

12.2. Claude 3 Model Family

Claude 3 family introduces multimodal capability and is released in three variants: Claude 3 Opus, Claude 3 Sonnet, and Claude 3 Haiku. These variants reflect trade-offs among reasoning capacity, latency, and computational efficiency. Opus targets complex analytical tasks, Sonnet balances performance and speed, and Haiku prioritizes low-latency deployment.

Claude 3 employs a Transformer architecture augmented with multimodal embeddings for joint text–image representation. Cross-modal attention enables integrated reasoning, while alignment layers constrain unsafe generation. The model supports context lengths of up to 200,000 tokens, enabling reasoning over large documents and multimodal inputs.

Training data comprise a mixture of public sources, licensed datasets, and curated internal content. Preprocessing includes normalization, deduplication, and safety filtering. Training objectives include next-token prediction for text and masked prediction for visual inputs, supported by large-scale

 doi:10.20944/preprints202601.1138.v1

parallelism and checkpointing. Fine-tuning incorporates task supervision and RLHF, with learning rates in the range of 1×10^{-5} to 2×10^{-5} and batch sizes between 128 and 512.

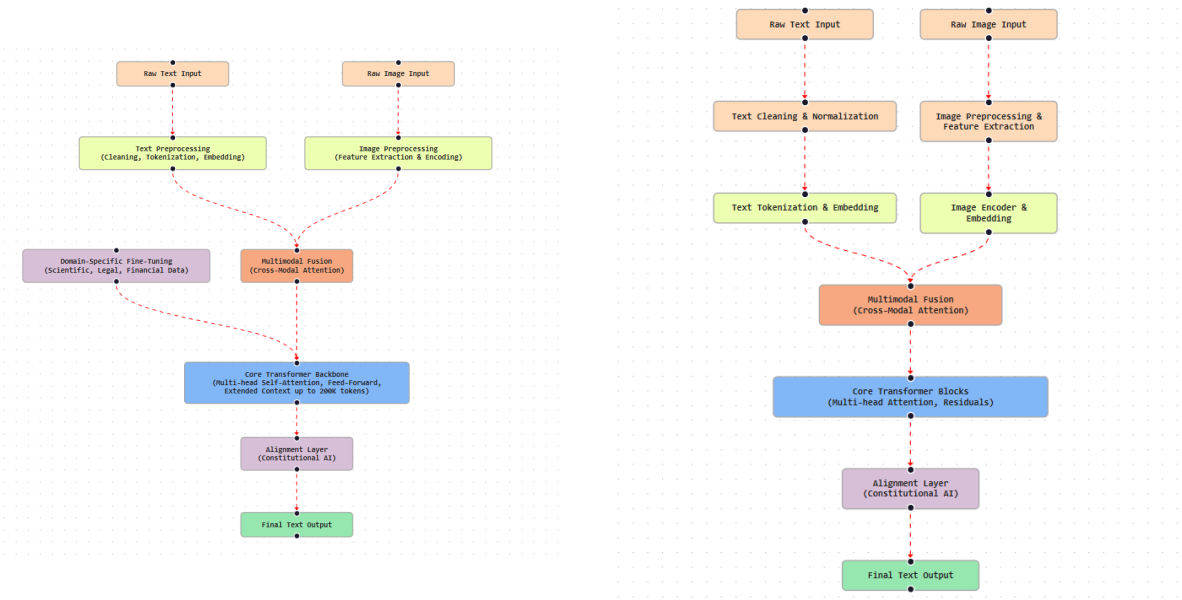

Figure 13. Variants within the Claude 3 family. The left panel shows Claude 3 Opus, optimized for high-capacity reasoning, while the right panel shows Claude 3 Sonnet, balancing performance and efficiency.

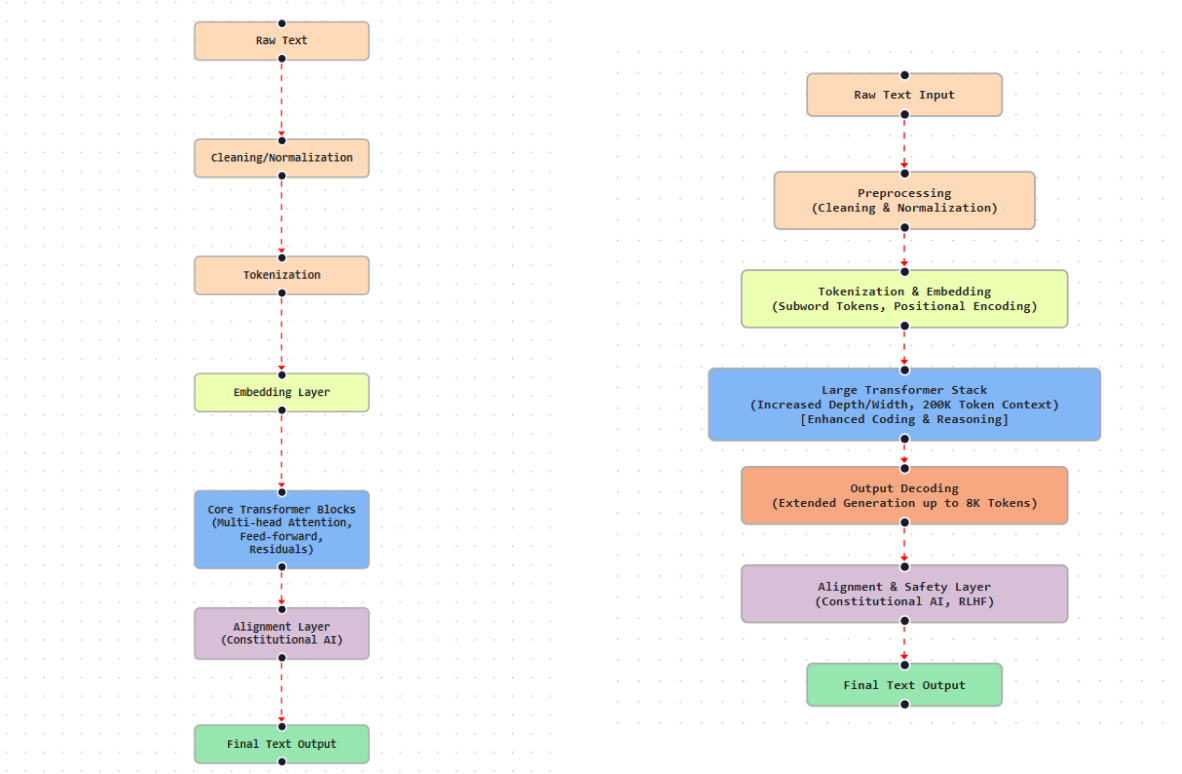

Figure 14. Lightweight variants in the Claude family. The left panel shows Claude 3 Haiku, while the right panel presents Claude 3.5 Haiku with improved efficiency and instruction adherence.

12.3. Claude 3.5 and Capabilities

Claude 3.5 extends the Claude 3 family with refinements targeting responsiveness, efficiency, and instruction-following performance. Claude 3.5 Haiku achieves a pass@1 score of 40.6% on SWE-bench and demonstrates strong performance on benchmarks such as MMLU and MATH [46]. Although

 doi:10.20944/preprints202601.1138.v1

optimized for text-centric workloads, the model is well suited for academic, educational, and business applications.

Claude 3 Sonnet shows strong multimodal reasoning, achieving competitive results on benchmarks including MathVista, ChartQA, and AI2D. It attains a 49% pass@1 score on SWE-bench Verified and exhibits high refusal accuracy on harmful prompts, indicating effective alignment.

12.4. Capabilities and Applications

Claude models support behavioral control over tone, structure, and response format, enabling flexible deployment across enterprise automation, research assistance, education, and creative tasks [13]. Combined with long-context reasoning and alignment-focused generation, Claude is well suited for information-dense and safety-sensitive domains.

12.5. Safety, Alignment, and Trust

Safety and alignment are central to Anthropic's development framework. Claude models employ Constitutional AI to enable self-correction guided by predefined ethical principles, reducing reliance on manual oversight [13]. Reinforcement Learning from AI Feedback (RLAIF) further refines alignment through AI-generated feedback. Content filtering, self-evaluation, and structured reasoning mechanisms mitigate bias, hallucination, and harmful generation, supporting robust and trustworthy deployment.

13. Falcon AI

Falcon AI is a family of large-scale Transformer-based language models developed by the Technology Innovation Institute (TII) under the Advanced Technology Research Council (ATRC) [18]. Falcon series emphasizes open accessibility, efficient large-scale training, and competitive performance, contributing to sovereign and open large language model development.

13.1. Falcon-7B

Falcon-7B adopts a causal decoder-only Transformer architecture optimized for autoregressive generation. The model employs RoPE and multiquery attention with grouped key–value projections to reduce inference-time memory overhead and improve throughput. Attention and MLP computations are executed in parallel within each Transformer block, reducing synchronization overhead. Bias-free linear layers and normalization operations support stable large-scale optimization.

The model is pretrained on approximately 1.5 trillion tokens from the RefinedWeb dataset [47], consisting of filtered web content and curated sources such as books and dialogues, without upsampling. Training is conducted on 384 NVIDIA A100 GPUs using three-dimensional parallelism and ZeRO optimizer sharding. Mixed-precision training with `bfloat16` and the AdamW optimizer (weight decay $= 0.1$) ensures computational efficiency. Parameter-efficient fine-tuning methods, including LoRA and LLaMA-Adapter, enable downstream adaptation with minimal resource overhead [48].

13.2. Falcon-40B

Falcon-40B scales the Falcon architecture to 40B parameters while retaining the decoder-only design, RoPE positional encoding, multigroup attention, and parallelized attention–MLP execution. The model is trained on a one-trillion-token subset of RefinedWeb with comprehensive filtering and deduplication [18]. Optimization uses AdamW with ZeRO sharding on A100 GPU clusters. For downstream tasks, Falcon-40B supports efficient adaptation through QLoRA, enabling fine-tuning with a small fraction of trainable parameters.

13.3. Falcon-180B

Falcon-180B further scales the architecture to 180B parameters while preserving the core design principles of earlier Falcon models. The model uses multigroup attention, RoPE, parallelized attention–MLP execution, and GeLU activations for efficiency. Training is performed on approximately 3.5 trillion

 doi:10.20944/preprints202601.1138.v1

tokens, primarily sourced from RefinedWeb, supplemented with curated academic, code, and book content. Optimization employs large-scale GPU parallelism, z-loss regularization, gradient clipping, and cosine learning-rate schedules to maintain stability [18].

14. Falcon 2 Series

The Falcon 2 series introduces architectural refinements, extended context handling, and multimodal capability to support broader generalization and scalability [49]. These models build upon the Falcon 1 lineage while improving efficiency, long-context reasoning, and modular integration. Figure 15 illustrates the core architectural components of the Falcon 2 11B models, highlighting the contrast between the language-only backbone and its multimodal extension for vision–language understanding.

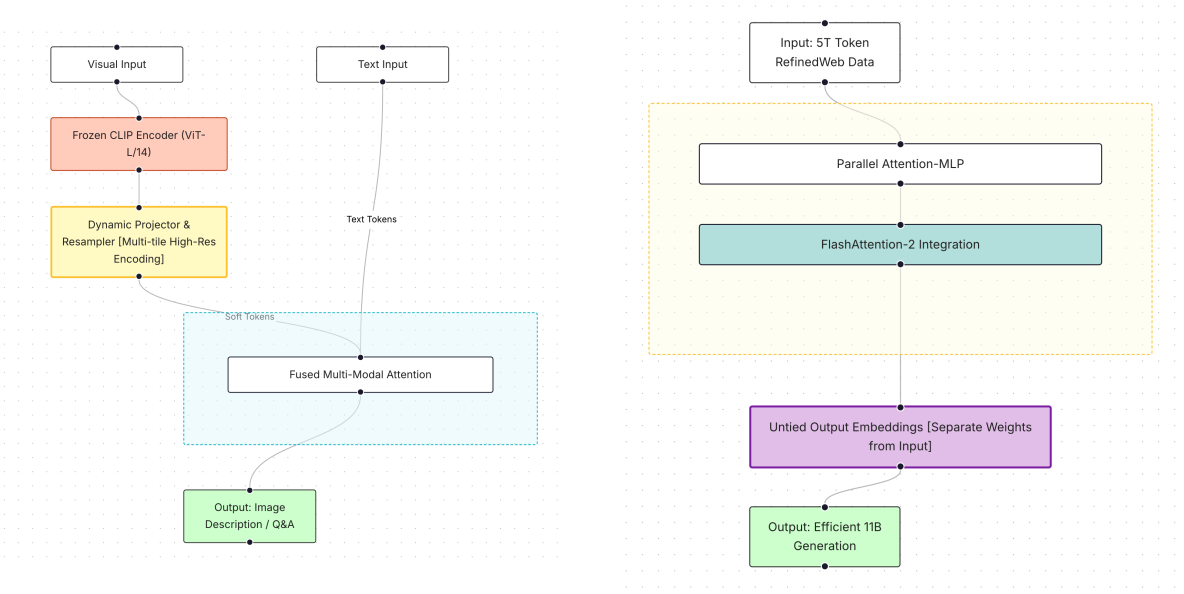

Figure 15. Architectural overview of the Falcon 2 11B model family. **Left:** The Falcon 2 11B Vision–Language Model (VLM) integrates a frozen CLIP ViT-L/14 visual encoder with a dynamic projector to fuse visual and text tokens via multimodal attention. **Right:** The Falcon 2 11B language model employs parallel attention–MLP blocks, FlashAttention-2, untied embeddings, and efficient large-scale text generation.

14.1. Falcon2-11B

Falcon2-11B is a decoder-only Transformer with 60 layers, a hidden size of 4096, and 32 attention heads. GQA with eight key–value heads improves memory efficiency. Attention and feed-forward layers are executed in parallel, reducing sequential dependencies and increasing throughput. FlashAttention-2 supports context lengths up to 8192 tokens, while modified RoPE scaling improves long-context modeling.

The model is trained on over five trillion tokens, primarily from RefinedWeb, with multilingual coverage and curated sources such as Reddit, Stack Exchange, and permissively licensed code. Optimization uses `bfloat16` precision, AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and z-loss regularization. Training follows a four-stage curriculum with progressively increasing sequence lengths and a final fine-tuning phase on high-quality domain-specific data, enabling instruction-following and dialogue capability.

14.2. Falcon2-11B VLM

Falcon2-11B VLM extends the base model with vision–language capability by integrating a frozen CLIP ViT-L/14 vision encoder. Image embeddings are projected into the language model embedding space via a lightweight feed-forward projector and concatenated with text embeddings. Training

 doi:10.20944/preprints202601.1138.v1

proceeds in two phases: an initial alignment stage where only the multimodal projector is trained on image–text pairs, followed by joint optimization of the language model and projector using multimodal instruction data. The vision encoder remains frozen throughout, preserving linguistic performance while enabling visual grounding.

15. Falcon Mamba 7B

Falcon Mamba 7B departs from attention-based architectures by replacing self-attention with a state-space model (SSM) layer known as Mamba [50]. This design maintains a recurrent hidden state, enabling linear-time sequence processing with near-constant memory usage, making it suitable for long-context inference.

The model comprises 64 layers with a hidden size of 4096, totaling approximately 7.27B parameters. Feed-forward layers use an expansion factor of two, RMSNorm is applied before and after key transformations, and input and output embeddings are untied to improve stability.

15.1. Data and Training

Falcon Mamba 7B is trained on approximately 5.8 trillion English-language tokens drawn from RefinedWeb, curated academic text, permissively licensed code, and mathematical datasets. Mathematical content is upsampled to compensate for limited availability. Training is conducted on 256 NVIDIA H100 GPUs using AdamW optimization and ZeRO parallelism. The learning rate follows a warmup–plateau–decay schedule, with batch sizes gradually increased to maintain stability.

Pretraining proceeds through multiple curriculum stages with increasing sequence lengths, followed by a final decay phase emphasizing high-quality curated data. No supervised instruction fine-tuning is applied prior to release, enabling flexible downstream adaptation. Both final and pre-decay checkpoints are publicly released.

16. Falcon Series 3

Falcon 3 series represents a recent advance in scalable and efficient language modeling. The series combines Transformer-based and state-space architectures to support long-context reasoning, multilingual coverage, and computational efficiency across a range of deployment settings [51]. Model variants span lightweight configurations for resource-constrained environments and higher-capacity systems intended for research and industrial use. Figure 16 presents two representative Falcon 3 architectures, contrasting the Transformer-based design with the Mamba state-space alternative for long-context modeling. Additional Falcon variants and scaling configurations are available through the interactive LLM Architecture Explorer at https://devisri-b.github.io/LLM-Architectures/.

 doi:10.20944/preprints202601.1138.v1

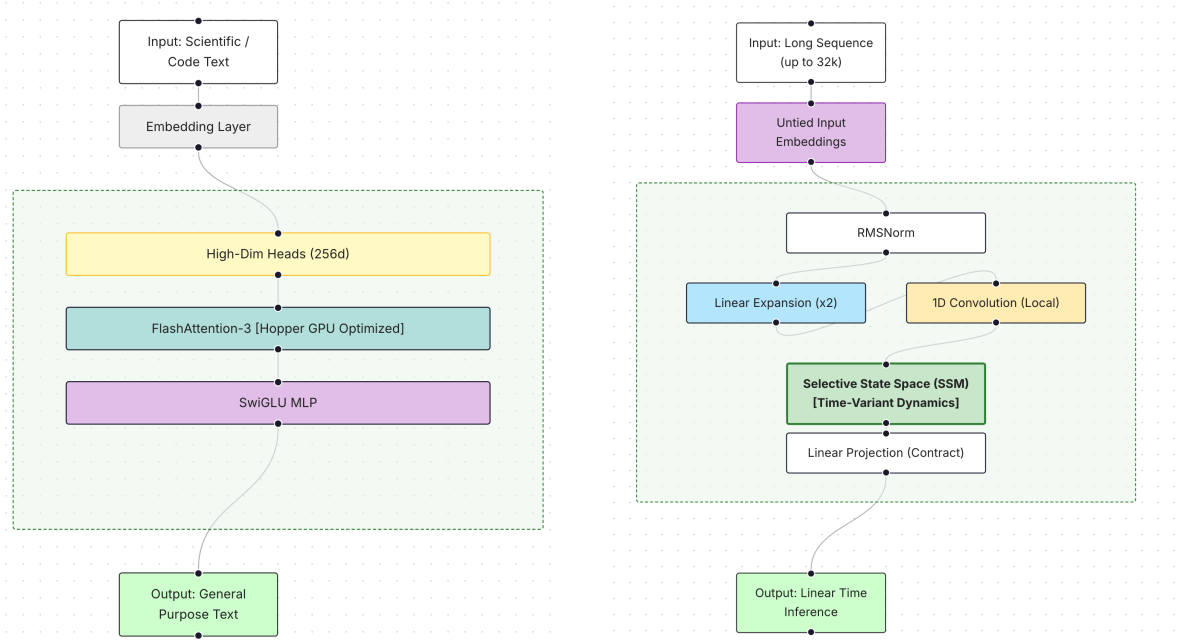

Figure 16. Representative architectures from the Falcon 3 model family. **Left:** Falcon 3 7B Transformer-based architecture, employing grouped-query attention (GQA), SwiGLU MLPs, RMSNorm, and extended context handling for strong general-purpose reasoning. **Right:** Falcon 3 Mamba 7B architecture, which replaces attention with selective state space models (SSMs) to enable linear-time inference for long-sequence modeling.

16.1. Falcon3-1B

Falcon3-1B is a decoder-only Transformer model with approximately one billion parameters and support for an 8,000-token context window. The architecture is LLaMA-compatible and uses a shared 131K-token vocabulary across the Falcon 3 family. Training is conducted on fewer than 100 gigatokens of curated web, code, STEM, and multilingual data. To improve efficiency, Falcon3-1B is trained using knowledge distillation from Falcon3-7B, enabling transfer of linguistic and reasoning capability with reduced computational cost. An instruction-tuned variant further refines performance for interactive and dialogue-oriented tasks.

16.2. Falcon3-3B-Base

Falcon3-3B-Base scales the Transformer architecture to three billion parameters across 22 layers with 256-dimensional attention heads and SwiGLU activation functions. The model supports a 32,000-token context window and retains compatibility with the shared 131K-token vocabulary. Pretraining is performed on fewer than 100 gigatokens of deduplicated and filtered data drawn from web, code, and scientific sources. Knowledge distillation from Falcon3-7B is incorporated through a combined cross-entropy and distillation loss. Fine-tuning uses supervised datasets and optional preference modeling to improve alignment with human expectations.

16.3. Falcon3-Mamba-7B-Base

Falcon3-Mamba-7B-Base adopts a state-space language model architecture with 64 layers and approximately seven billion parameters. The model employs a reduced vocabulary of 65K tokens, SwiGLU activations, and supports sequence lengths up to 32,000 tokens. State-space layers enable efficient modeling of long-range dependencies with linear scaling in memory and computation. Pretraining emphasizes multilingual and STEM-focused corpora and is conducted over 1.5 trillion tokens using large-scale GPU infrastructure. Fine-tuning targets reasoning and mathematical tasks, yielding improved performance on benchmarks such as BBH and MATH-Lvl5.

 doi:10.20944/preprints202601.1138.v1

16.4. Falcon3-7B-Base

Falcon3-7B-Base follows a decoder-only Transformer design with seven billion parameters distributed across 28 layers and 256-dimensional attention heads. The model supports a 32,000-token context window and uses the shared 131K-token vocabulary. FlashAttention-3 is integrated to improve training efficiency. Pretraining is performed on approximately 14 trillion tokens drawn from multilingual web data, source code, and STEM texts. Fine-tuning focuses on reasoning, mathematics, and code benchmarks, including ARC Challenge, BBH, GSM8K, MBPP, and Multipl-E.

16.5. Falcon3-10B-Base

Falcon3-10B-Base extends the Falcon 3 Transformer architecture to ten billion parameters by increasing depth to 40 layers. The model maintains support for the 131K-token vocabulary and 32,000-token context length. Training emphasizes structured STEM knowledge and multilingual reasoning data and is conducted over approximately two trillion tokens. Fine-tuning targets advanced reasoning and coding tasks, with reported improvements on benchmarks including MATH-Lvl5, GSM8K, MMLU/MMLU-PRO, MBPP, and Multipl-E.

16.6. Deployment and Privacy Considerations

Deployment of Falcon 3 models incorporates security and compliance measures such as encrypted data transfer, access control, and adherence to data protection regulations including the General Data Protection Regulation (GDPR) [52]. The models support applications in customer support automation, analytics, and intelligent monitoring, with integrated diagnostics and telemetry for performance monitoring and adaptive tuning.

Despite these safeguards, open questions remain regarding long-term data retention, cross-border data transfer, and transparency of telemetry mechanisms. Ongoing discussion continues around user consent, cookie policies, and data governance practices to support responsible deployment [52].

17. DeepSeek-R1

DeepSeek AI, founded in 2023 by Liang Wenfeng, has positioned itself as a competitive open-model developer with emphasis on reasoning-centric performance and computational efficiency [53,54]. DeepSeek-R1 is a flagship reasoning-focused model designed through a reinforcement-learning–centric training paradigm that departs from conventional supervised pretraining pipelines.

17.1. Model Architecture and Training Pipeline

DeepSeek-R1 is optimized for advanced reasoning through a multi-stage training pipeline integrating reinforcement learning (RL), synthetic data generation, and supervised fine-tuning [55]. The initial variant, DeepSeek-R1-Zero, is trained exclusively using reinforcement learning without supervised pretraining. This model employs Group Relative Policy Optimization (GRPO), which optimizes group-level reasoning improvements rather than token-level likelihoods. While this approach yields strong emergent reasoning, early outputs exhibit limited clarity and stylistic instability.

To address these limitations, a cold-start fine-tuning stage is introduced using curated high-quality reasoning examples, improving coherence and linguistic structure. A subsequent RL phase further refines reasoning depth using reward functions emphasizing logical correctness and conciseness. Synthetic reasoning trajectories are then generated and filtered via rejection sampling, reducing dependence on human annotation. These synthetic samples are used in an additional supervised fine-tuning stage to improve generalization. A final RL phase aligns outputs with human preferences, reducing hallucinations and balancing fluency with accuracy. The overall training pipeline is illustrated in Figure 17.

 doi:10.20944/preprints202601.1138.v1

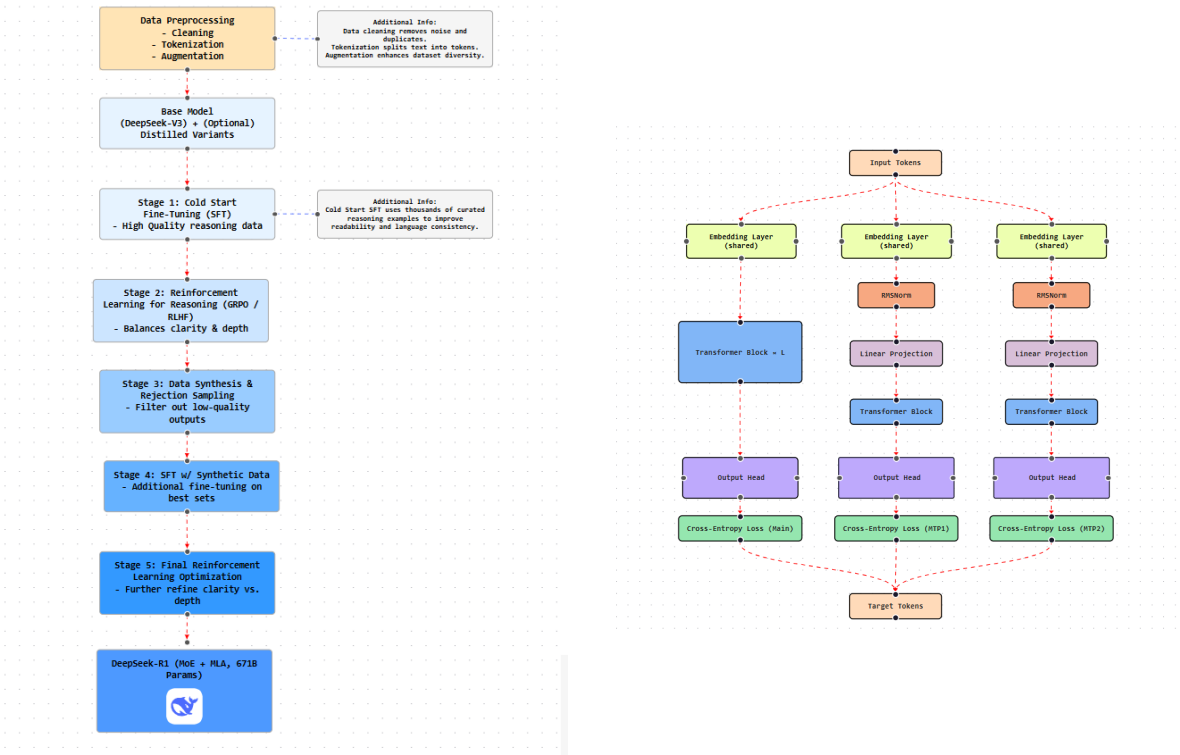

Figure 17. Comparison of DeepSeek model variants. The left panel depicts the reinforcement-learning–centric training pipeline of DeepSeek-R1, while the right panel shows the decoder-only Transformer architecture of DeepSeek-V3 with Multi-Head Latent Attention and long-context optimization.

17.2. Model Evolution and Performance

DeepSeek-R1 shows strong performance on tasks requiring logical inference, mathematical reasoning, and multi-hop question answering. Knowledge distillation is applied to transfer reasoning capability to smaller models for resource-constrained deployment. Table 3 summarizes key differences between DeepSeek-R1-Zero and the final DeepSeek-R1 model.

Table 3. Comparison between DeepSeek-R1-Zero and DeepSeek-R1.

Feature	**DeepSeek-R1-Zero**	**DeepSeek-R1**
Training Paradigm	RL-only (GRPO)	Hybrid RL + SFT + Synthetic Data
Reasoning Quality	High but unstable	Optimized and consistent
Language Coherence	Limited	Structured and fluent
Data Sources	RL-generated	Human-labeled + synthetic
Output Style	Variable	Normalized

17.3. Data Processing and Optimization

Data preprocessing follows rigorous quality-control procedures [56]. Duplicate and low-quality samples are removed using MinHash and TF–IDF filtering, while normalization standardizes punctuation and formatting. Subword tokenization (e.g., BPE, WordPiece) improves lexical consistency. Synthetic reasoning data is generated through RL and diversified via paraphrasing. Reward-based filtering prioritizes logically consistent samples.

 doi:10.20944/preprints202601.1138.v1

17.4. MoEs and Attention Design

DeepSeek-R1 employs a MoEs architecture with 671B total parameters, of which 37B are activated per token [57,58]. The model supports a 128K token context window, enabling document-level reasoning. MLA replaces standard MHA, compressing key–value representations via low-rank latent factors and reducing computation by approximately 5–13%. MoE routing mechanism balances scalability and efficiency while supporting multi-step reasoning strategies that favor interpretability over brevity.

17.5. Generation Characteristics

Dynamic expert routing enables DeepSeek-R1 to adapt generation behavior across reasoning, summarization, and open-ended tasks. While multi-step reasoning increases token usage, it improves transparency and logical reliability, supporting applications requiring interpretable reasoning traces.

18. DeepSeek-V3

DeepSeek-V3 is the latest flagship model from DeepSeek AI, designed to deliver strong long-context reasoning and computational efficiency at scale [53,54]. The model integrates architectural and training innovations to compete with leading proprietary systems while maintaining cost efficiency.

18.1. Architectural Innovations

DeepSeek-V3 introduces several architectural refinements (Figure 17). MLA compresses key–value pairs via low-rank decomposition and separates contextual and positional components, reducing memory usage and accelerating inference. The model adopts a MoEs architecture with 671B total parameters and 37B active per token, using dynamic routing without auxiliary load-balancing losses. Multi-Token Prediction (MTP) enables parallel token generation, improving both training throughput and inference latency.

18.2. Training Methodology

DeepSeek-V3 is trained on 14.8 trillion tokens spanning code, mathematics, scientific text, and multilingual data [59]. Optimization strategies include FP8 mixed-precision training, DualPipe pipeline parallelism, and bias-based expert routing. Training required approximately 2.66M H800 GPU hours at a reported cost of $5.576M. Fine-tuning integrates knowledge distillation from DeepSeek-R1, supervised instruction tuning, and RLHF using GRPO.

18.3. Data Pipeline

Preprocessing emphasizes quality, diversity, and bias mitigation:

- Enhanced coverage of programming, mathematics, and multilingual content.
- Byte-level BPE with a 128K vocabulary and optimized pretokenization.
- Fill-in-the-Middle (FIM) training to improve interpolation.
- Token boundary randomization to reduce few-shot bias.

18.4. Performance and Capabilities

DeepSeek-V3 achieves near state-of-the-art performance across reasoning, code generation, and long-context QA. It performs comparably to GPT-4o and Claude 3.5 Sonnet on benchmarks such as MMLU-Redux and GPQA-Diamond, attains a HumanEval-Mul Pass@1 of 82.6, and demonstrates strong robustness on 128K-context QA tasks. Table **??** summarizes key feature-level comparisons.

18.5. Safety and Alignment

DeepSeek-V3 integrates layered safety mechanisms [60], including dual-stage content filtering, RLHF-based bias mitigation, enterprise-grade security controls, and real-time adversarial testing. These measures support robust deployment while maintaining transparency and efficiency.

 doi:10.20944/preprints202601.1138.v1

Overall, DeepSeek-V3 provides a scalable and cost-efficient open alternative to leading proprietary models, combining MLA, MTP, and MoE routing to deliver strong reasoning, multilingual performance, and long-context capability.

19. Qwen AI

Qwen is a family of LLMs developed by Alibaba Cloud, designed to support a wide range of language-centric tasks including text generation, translation, code synthesis, and mathematical reasoning [61]. The models are based on a decoder-only Transformer architecture and emphasize architectural efficiency, stable optimization, and long-context capability within an open-weight framework.

19.1. Architecture and Core Design

Qwen adopts a decoder-only Transformer architecture augmented with design choices commonly associated with recent large-scale language models. RoPE is used to encode relative positional information, enabling effective generalization to long sequences. Root Mean Square Layer Normalization (RMSNorm) is employed to improve optimization stability and reduce computational overhead, while SwiGLU activation functions enhance gradient flow in deep networks. Input and output embeddings are untied to increase representational flexibility.

Most linear layers are implemented without bias terms to improve efficiency, with exceptions in query–key–value projections. Attention computation is accelerated using FlashAttention. To support long-context extrapolation, Qwen incorporates LogN-scaled attention and NTK-aware interpolation, which stabilize attention behavior beyond the original training context length. Figure 18 presents an overview of the model architecture.

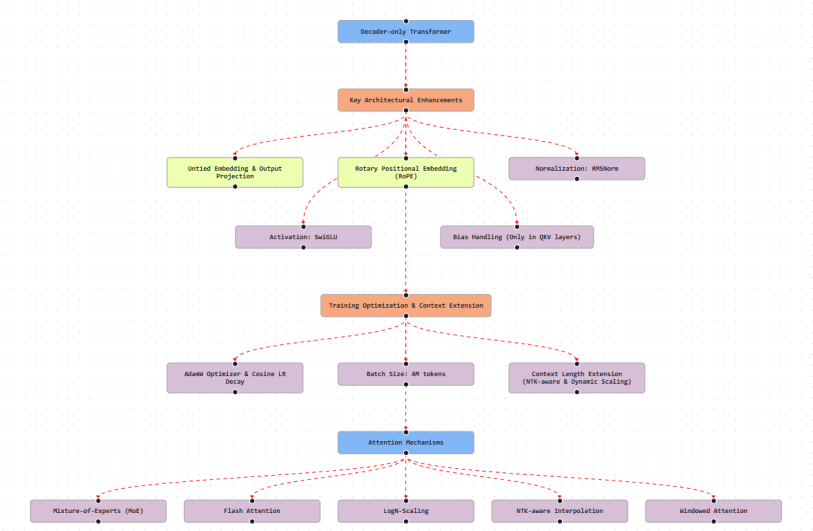

Figure 18. Overview of the Qwen architecture, highlighting its decoder-only Transformer backbone, attention optimizations, and mechanisms for efficient long-context modeling.

19.2. Training Strategy and Data Pipeline

Qwen models are trained using large-batch, mixed-precision optimization, typically with the AdamW optimizer and cosine learning-rate schedules with warm-up. This setup supports stable convergence at large parameter scales and extended sequence lengths.

The training corpus comprises trillions of tokens drawn from multilingual web content, books, academic sources, and open-source code repositories. Preprocessing includes deduplication, language filtering, and removal of low-quality or harmful content. Instruction-style datasets are introduced in later training stages to improve instruction adherence and downstream task performance.

 doi:10.20944/preprints202601.1138.v1

19.3. Model Variants and Configurations

Qwen family includes several variants tailored to different use cases. Base pretrained models (Qwen-Base) provide general-purpose language representations, while instruction-tuned variants (Qwen-Chat) are aligned using supervised fine-tuning and RLHF. Domain-specialized models, such as Code-Qwen and Math-Qwen, focus on programming-related tasks and mathematical reasoning.

Models are released in multiple parameter scales, including 1.8B, 7B, and 14B configurations. All variants share common architectural components, including RoPE, RMSNorm, SwiGLU activations, and cosine learning-rate schedules, while differing in depth, hidden size, and attention head counts to balance performance and computational cost.

19.4. Capabilities and Performance

Qwen demonstrates competitive performance across a range of language tasks. In text generation, it supports controlled decoding strategies such as top-k, top-p, and beam search, producing coherent and contextually appropriate outputs. Multilingual training enables effective translation across both high-resource and low-resource languages.

In code generation, Code-Qwen performs well on benchmarks such as HumanEval and MBPP, supporting multiple programming languages including Python, Java, and C++. Qwen also achieves strong results on summarization and question-answering tasks, leveraging long-context modeling and instruction tuning to produce concise summaries and accurate responses.

19.5. Safety, Privacy, and Bias

Qwen incorporates safeguards to support responsible deployment, including privacy-aware training practices, data encryption, and compliance with data protection regulations such as GDPR. Alignment mechanisms, including RLHF and fairness-aware data filtering, are applied to reduce harmful or biased outputs.

Despite these measures, challenges remain. The model can exhibit susceptibility to prompt injection and residual bias, particularly in multilingual or adversarial settings. These limitations highlight the need for continued robustness evaluation, bias auditing, and refinement to support reliable real-world deployment.

20. Comparative Analysis of LLMs

To synthesis the architectural and methodological differences discussed throughout this survey, this section presents a comparative analysis of contemporary LLMs across key dimensions, including training and alignment strategies, context length and tokenization, benchmark performance, multimodal capability, and safety mechanisms. The comparison highlights both shared design foundations and divergent approaches that influence scalability, efficiency, and deployment readiness.

Table 4 summarizes the defining characteristics of major LLM families. Across models, supervised fine-tuning (SFT) combined with RLHF has emerged as a common alignment strategy. Proprietary systems such as GPT-4 and Claude 3 extend this paradigm through additional alignment mechanisms, including PPO, Constitutional AI, and RLAIF. In contrast, open and semi-open models such as Falcon and DeepSeek emphasize efficiency-oriented techniques, including gradient clipping, z-loss regularization, rejection sampling, and selective knowledge distillation.

Supported context length represents a major axis of differentiation. Earlier architectures, including LLaMA 2, typically operate within 4K-token limits, whereas more recent models extend context capacity through optimized attention and routing mechanisms. GPT-4o and DeepSeek support contexts on the order of tens of thousands of tokens, while Gemini 1.5 introduces a substantial shift toward extreme long-context modeling, enabling up to one million tokens through memory-efficient attention, expert routing, and hybrid tokenization. These advances facilitate reasoning over long documents, codebases, and structured multimodal inputs.

 doi:10.20944/preprints202601.1138.v1

Multimodal capability further distinguishes modern LLMs. GPT-4o and Gemini provide unified processing across text, image, audio, and video, supporting cross-modal reasoning and interactive applications. Claude 3 introduces partial multimodality through vision–language integration, while Qwen and DeepSeek remain primarily text-centric but exhibit strong performance in specialized domains such as multilingual question answering, structured reasoning, and code generation.

Overall, the comparative analysis reveals convergence toward Transformer-based foundations alongside increasing diversification in scaling strategies, alignment methods, and deployment objectives. Proprietary models tend to prioritize multimodal integration and extended-context reasoning, whereas open and semi-open models explore efficiency-driven architectures and targeted specialization. These complementary trajectories reflect the evolving landscape of LLM development and the trade-offs shaping next-generation AI systems.

Table 4. Comparative analysis of major LLM families.

Model	Training Techniques	Context Length / Tokenization	Performance Benchmarks	Multimodal Capabilities	Safety & Alignment
GPT-4 / GPT-4o	SFT, RLHF, PPO, feedback loops	128K (GPT-4o), BPE	SOTA on MMLU, HumanEval, DROP	Text, image, audio, video	RLHF, moderation APIs, content filters, human evaluation
GPT-O1 (preview/mini)	SFT, RLHF, chain-of-thought	8K–32K, BPE	Logical reasoning, code pass@1	Text	Reward models, jailbreak resistance, policy optimization
LLaMA 2	SFT, RLHF, Ghost Attention	4K, RoPE	BLEU/COMET, QA, summarization	Text	Safety reward models, RLHF, ToxiGen, TruthfulQA
Gemini 1.5	SFT, RLHF, MoE routing	Up to 1M, hybrid tokenizer	HumanEval, MMLU, Natural2Code	Text, image, audio, video, code	Expert gating, moderation filters, on-device privacy
Claude 2 / Claude 3	SFT, RLHF, Constitutional AI	100K–200K, BPE	SWE-bench, MathVista, ChartQA	Text, image (Claude 3)	Self-critique, RLAIF, red teaming
Falcon Series (180B/2/3)	Gradient clipping, z-loss, distillation	4K–32K, RoPE	BBH, GSM8K, ARC, MBPP	Text (VLM in Falcon2)	Safety classifiers, LoRA-based alignment
DeepSeek (R1 / V3)	RL-only (Zero), SFT + RL + synthetic data	32K–128K, RoPE / MLA	Logical reasoning, math, CoT QA	Text	Reward filtering, rejection sampling
Qwen AI	SFT, RLHF, preference optimization	32K+, RoPE, untied embeddings	Multilingual QA, code, summarization	Text	Fairness filtering, RLHF, bias mitigation

21. Discussion

The rapid evolution of LLMs reflects the convergence of advances in architecture, large-scale optimization, alignment methodologies, and deployment practices. Building on the comparative analysis and architectural taxonomy presented in this survey, several overarching trends and trade-offs emerge that characterize the current LLM landscape and shape future research directions.

21.1. Trends in Scaling, Architecture, and Safety

Recent LLM development is marked by sustained growth in model scale and supported context length. Models such as GPT-4 and Gemini 1.5 extend parameter counts into the hundreds of billions and support increasingly long contexts, enabling document-level reasoning and multimodal integration. Accommodating this scale has driven the adoption of modular and efficiency-oriented architectures, including Mixture-of-Experts routing, Multi-Head Latent Attention, and memory-efficient attention mechanisms such as FlashAttention and Grouped Query Attention.

Alongside architectural scaling, safety and alignment have become central design considerations. RLHF is now a standard component of alignment pipelines, complemented by techniques such as Constitutional AI, reward modeling, and self-critique mechanisms. Models such as Claude 3 and GPT-O1 incorporate explicit safeguards and resistance to adversarial prompting, reflecting a shift toward embedding alignment directly into training and deployment rather than relying on post-processing controls.

 doi:10.20944/preprints202601.1138.v1

21.2. Trade-Offs Between Model Complexity and Performance

Performance gains achieved by large-scale models are accompanied by substantial computational, financial, and operational costs. High-capacity models such as GPT-4 and Claude Opus require extensive training resources and specialized infrastructure, which can limit deployment flexibility and accessibility.

In contrast, efficiency-oriented models, including Qwen AI, Falcon 3 variants, and Claude Haiku, emphasize deployability and cost efficiency. While these models may not consistently reach the peak performance of larger systems, they achieve competitive results in targeted tasks such as code generation, summarization, and multilingual question answering. This highlights a key practical insight: model suitability is highly application-dependent, and compact LLMs play an essential role in enabling broader adoption, particularly in resource-constrained and on-device environments.

21.3. Open-Source and Proprietary Development Paradigms

LLM ecosystem is further shaped by the distinction between proprietary and open development paradigms. Proprietary models such as GPT-4, Claude, and Gemini typically lead in benchmark performance, multimodal capability, and alignment sophistication, but offer limited architectural transparency and restricted access. While this approach supports controlled deployment and rapid iteration, it constrains reproducibility and independent evaluation.

Open and semi-open initiatives, including LLaMA 2, Falcon, DeepSeek, and Qwen AI, prioritize transparency through the release of model weights, training details, and evaluation protocols. This openness enables community-driven experimentation, independent safety assessment, and rapid downstream innovation, although such models may lag behind proprietary systems in alignment refinement and large-scale multimodal integration due to resource limitations.

Collectively, these paradigms form a complementary ecosystem. Proprietary models advance the frontier of performance and alignment, while open-source models promote accessibility, accountability, and reproducibility. Their continued interaction is likely to remain a defining factor in the evolution of LLMs, influencing how future systems balance capability, efficiency, safety, and societal impact.

22. Future Directions

As research on LLMs continues to advance, progress will increasingly depend on considerations beyond model scale alone. While parameter growth and extended context windows have driven recent improvements, emerging challenges underscore the need for greater transparency, efficiency, inclusivity, and accountability. This section outlines key directions likely to shape the next generation of foundation models.

22.1. Transparent and Standardized Evaluation

Despite strong reported performance, evaluation practices for LLMs often rely on proprietary datasets or partially disclosed protocols, limiting reproducibility and cross-model comparability. Future work should prioritize transparent and standardized evaluation frameworks that assess a broad spectrum of capabilities, including reasoning, code generation, safety, factual consistency, and multimodal understanding. Existing benchmarks such as HELM, BIG-Bench, and MMLU provide useful starting points but require extension to better capture long-context reasoning, multilingual robustness, and domain-specific generalization. Standardized evaluation is essential for meaningful progress assessment and responsible deployment.

22.2. Cross-Lingual Generalization and Low-Resource Settings

Many contemporary LLMs remain disproportionately optimized for high-resource languages, resulting in uneven performance across linguistic and cultural contexts. Addressing this imbalance represents both a technical challenge and an ethical priority. Promising directions include multilingual data augmentation, language-specific adapters, and transfer learning from multilingual encoders.

 doi:10.20944/preprints202601.1138.v1

Community-driven dataset creation and evaluation will also be critical for supporting underrepresented languages, dialects, and scripts, enabling more equitable access to language technologies.

22.3. Long-Context Reasoning and Memory-Augmented Models

Supporting effective long-context reasoning remains a central challenge for applications such as legal analysis, scientific synthesis, and sustained dialogue. Although recent models demonstrate context windows exceeding 100K tokens, these capabilities incur substantial computational and memory costs. Future research should explore architectural alternatives, including memory-augmented Transformers, state-space models, and hybrid attention–memory designs. Such approaches may enable persistent reasoning over extended inputs while maintaining efficiency and scalability.

22.4. Governance and Responsible Deployment

As LLMs are increasingly deployed in high-impact domains, robust governance and accountability mechanisms become essential. Developers and deployers should adopt standardized transparency practices, including Model Cards and Data Sheets, alongside systematic bias audits and red-teaming evaluations. Beyond technical safeguards, coordinated regulatory frameworks are needed to address misuse, accountability, and unequal access. Establishing shared governance norms that balance innovation with societal responsibility will be critical for ensuring trustworthy and ethical LLM deployment.

23. Conclusions

This survey presented a comprehensive analysis of contemporary LLMs, synthesizing architectural design choices, training and alignment strategies, multimodal capabilities, and performance characteristics across major model families, including GPT, Claude, LLaMA, Gemini, Falcon, DeepSeek, and Qwen AI. Through a unified taxonomy and comparative analysis, the study demonstrates that while continued scaling remains relevant, the primary differentiators of modern LLMs increasingly lie in efficiency-oriented architectures, long-context modeling, multimodal integration, and alignment mechanisms for safe deployment.

The analysis highlights the complementary roles of proprietary and open development paradigms within the LLM ecosystem. Proprietary models typically lead in multimodal capability, alignment refinement, and large-context reasoning, while open and semi-open initiatives contribute transparency, reproducibility, and adaptability, supporting community-driven innovation and broader accessibility. Together, these paradigms form an interdependent ecosystem that accelerates technical progress while expanding practical adoption.

For researchers, this survey provides a structured framework for understanding architectural trade-offs, alignment methodologies, and emerging design patterns across LLM families. For practitioners, the comparative insights support informed model selection based on application needs, computational constraints, and safety considerations. For policymakers and stakeholders, the findings underscore the growing importance of transparency, inclusivity, and accountability in large-scale AI development.

Looking ahead, the evolution of LLMs will require a shift from scale-centric advancement toward holistic design approaches that prioritize interpretability, fairness, efficiency, and responsible governance alongside performance. Achieving this balance will depend on sustained collaboration across academia, industry, and civil society to ensure that future LLMs remain both technologically impactful and socially beneficial.

References

1. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. Language models are unsupervised multitask learners. *OpenAI blog* **2019**, *1*, 9.

 doi:10.20944/preprints202601.1138.v1

2. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; 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.
3. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Lample, G.; et al. LLaMA 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288* **2023**.
4. Rane, A.; Choudhary, P.; Rane, P. Gemini 1.5 Technical Overview. https://arxiv.org/abs/2403.05530, 2024.
5. Anthropic. Claude 3 Model Family. *Anthropic Blog* **2024**.
6. AI, D. DeepSeek-R1: Reinforcement Learning Powered Reasoning. https://www.deepseek.com/blog/deepseek-r1, 2024.
7. AI, D. DeepSeek-V3: Multimodal Capabilities and Efficient Memory Use. https://www.deepseek.com/blog/deepseek-v3, 2024.
8. Cloud, A. Qwen: Language Models for Multilingual AI. https://www.alibabacloud.com/help/en/model-studio/what-is-qwen-llm, 2024.
9. (TII), T.I.I. Falcon: Open-Source Large Language Models. *TII Technical Report* **2023**.
10. OpenAI. GPT-4o: OpenAI's Multimodal Flagship. https://openai.com/index/hello-gpt-4o/, 2024.
11. 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.
12. OpenAI. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774* **2023**.
13. Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Kaplan, J.; et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073* **2022**.
14. Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* **2025**.
15. Bommasani, R.; Hudson, D.A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M.S.; Bohg, J.; Bosselut, A.; Brunskill, E.; et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* **2021**.
16. 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. *arXiv preprint arXiv:2203.02155* **2022**.
17. 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.
18. Almazrouei, E.; Alobeidli, H.; Alshamsi, A.; Cappelli, A.; Cojocaru, R.; Hesslow, D.; Launay, J.; Malartic, Q.; Mazzotta, D.; Noune, B.; et al. The Falcon Series of Open Language Models. *ArXiv (Cornell University)* **2023**. https://doi.org/10.48550/arxiv.2311.16867.
19. Su, J.; Lu, Y.; Pan, S.; Murtadha, A.; Wen, B.; Liu, Y. RoFormer: Enhanced Transformer with Rotary Position Embedding **2021**. https://doi.org/2104.09864.
20. Zhang, B.; Sennrich, R. Root mean square layer normalization. *Advances in Neural Information Processing Systems* **2019**, *32*.
21. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* **2017**.
22. Roumeliotis, K.I.; Tselikas, N.D. Chatgpt and open-ai models: A preliminary review. *Future Internet* **2023**, *15*, 192.
23. Achiam, J.; et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* **2023**.
24. OpenAI. GPT-4 Research. https://openai.com/index/gpt-4-research/, 2023.
25. OpenAI. GPT-4o System Card. https://openai.com/index/gpt-4o-system-card/, 2024.
26. OpenAI. OpenAI O1 System Card. https://cdn.openai.com/o1-system-card.pdf#page=16, 2024. Accessed: 2024-10-30.
27. Bhati, D.; Neha, F.; Guercio, A.; Amiruzzaman, M.; Kasturiarachi, A.B., Diffusion Model and Generative AI for Images. In *A Beginner's Guide to Generative AI: An Introductory Path to Diffusion Models, ChatGPT, and LLMs*; Springer Nature Switzerland: Cham, 2026; pp. 135–184. https://doi.org/10.1007/978-3-031-84724-0_6.
28. Meta. Introducing LLaMA: A foundational, 65-billion-parameter large language model. https://ai.meta.com/blog/large-language-model-llama-meta-ai/, 2023.
29. Meta. Meta and Microsoft Introduce the Next Generation of Llama. https://ai.meta.com/blog/llama-2/, 2023.
30. Portakal, E. LLaMA 2 Use Cases. https://textcortex.com/post/llama-2-use-cases, 2023.

 doi:10.20944/preprints202601.1138.v1

31. Xu, H.; Kim, Y.J.; Sharaf, A.; Awadalla, H.H. Advanced Language Model-based Translator (ALMA): A Fine-tuning Approach for Translation Using LLaMA-2. *arXiv preprint arXiv:2309.11674* **2023**.
32. Rozière, B.; Gehring, J.; Gloeckle, F.; Sootla, S.; Gat, I.; Tan, X.E.; Adi, Y.; Liu, J.; Sauvestre, R.; Remez, T.; et al. Code Llama: Open Foundation Models for Code. *arXiv preprint arXiv:2308.12950* **2023**.
33. Ahmed, W. Meta's Llama 2 Model & Its Capabilities. https://www.linkedin.com/pulse/metas-llama-2-model-its-capabilities-waqas-ala1f/, 2023.
34. Meta. Code Llama: An AI model for code generation and discussion. https://about.fb.com/news/2023/08/code-llama-ai-for-coding/, 2024.
35. Schmid, P.; Sanseviero, O.; Cuenca, P.; Tunstall, L.; von Werra, L.; Ben Allal, L.; Zucker, A.; Gante, J. Code Llama: Llama 2 learns to code. https://huggingface.co/blog/codellama, 2023.
36. Van Veen, D.; Van Udekemeier, L.; Delbrouck, J.B.; Aali, A.; Bluethgen, C.; et al. Clinical Text Summarization: Adapting Large Language Models Can Outperform Human Experts. *Nature Medicine* **2023**.
37. Google. Introducing Gemini: Google DeepMind's Next-Generation AI. https://blog.google/technology/ai/google-gemini-ai/#sundar-note, 2023.
38. Unite.ai. Google's Multimodal AI Gemini: A technical deep dive. https://www.unite.ai/googles-multimodal-ai-gemini-a-technical-deep-dive/, 2023.
39. Google Cloud. Essentials of Gemini: The new era of AI. https://medium.com/google-cloud/essentials-of-gemini-the-new-era-of-ai-efca53293341, 2023.
40. Swipe Insight. Google's Gemini training data sources: A deep dive into tech companies' transparency. https://web.swipeinsight.app/posts/google-s-gemini-training-data-sources-a-deep-dive-into-tech-companies-transparency-5846, 2023.
41. Islam, R.; Ahmed, I. Gemini: The most powerful LLM—Myth or truth. In Proceedings of the Proceedings of the 5th Information Communication Technologies Conference, 2024. https://doi.org/10.1109/ICTC61510.2024.10602253.
42. Pande, A.; Patil, R.; Mukkemwar, R.; Panchal, R.; Bhoite, S. Comprehensive study of Google Gemini and text generating models: Understanding capabilities and performance. *Grenze International Journal of Engineering and Technology* **2024**. June Issue.
43. Evonence. Gemini: The unrivaled platform for language translation. https://www.evonence.com/blog/gemini-the-unrivaled-platform-for-language-translation, 2024.
44. Anthropic. Model Card and Evaluations for Claude Models. https://www-cdn.anthropic.com/bd2a28d2535bfb0494cc8e2a3bf135d2e7523226/Model-Card-Claude-2.pdf, 2023.
45. Bai, Y.; et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862* **2022**.
46. Anthropic. The Claude 3 Model Family: Opus, Sonnet, Haiku. https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf, 2024.
47. Penedo, G.; et al. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116* **2023**.
48. Tian, X. Fine-tuning Falcon LLM 7B/40B. Lambda.ai Blog, 2023. https://lambda.ai/blog/fine-tuning-falcon-llm-7b/40b.
49. Malartic, Q.; et al. Falcon2-11b technical report. *arXiv preprint arXiv:2407.14885* **2024**.
50. Zuo, J.; et al. Falcon mamba: The first competitive attention-free 7b language model. *arXiv preprint arXiv:2410.05355* **2024**.
51. Team, F.L. The Falcon 3 Family of Open Models, 2024.
52. Stattelman, M. FALCONS.AI. https://falcons.ai/privacypolicy.html, 2021.
53. Wenfeng, L. DeepSeek AI founder Liang Wenfeng: The entrepreneur behind China's AI ambitions. https://apnews.com/article/deepseek-founder-liang-wenfeng-china-ai-0673d5c39d90108189cc31b88d85b9f8, 2024.
54. Neha, F.; Bhati, D. A Survey of DeepSeek Models. *Authorea Preprints*.
55. Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; He, Y. Deepseek-R1: Incentivizing Reasoning Capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948* **2025**.
56. Mumuni, A.; Mumuni, F. Automated data processing and feature engineering for deep learning and big data applications: A survey. *arXiv preprint arXiv:2403.11395* **2024**.
57. NVIDIA. DeepSeek-R1 Model Card. https://build.nvidia.com/deepseek-ai/deepseek-r1/modelcard, 2025.
58. AWS. DeepSeek-R1 Model Now Available in Amazon Bedrock Marketplace and Amazon SageMaker JumpStart. https://aws.amazon.com/blogs/machine-learning/deepseek-r1-model-now-available-in-amazon-bedrock-marketplace-and-amazon-sagemaker-jumpstart/, 2025.

 doi:10.20944/preprints202601.1138.v1

59. Lu, H.; Liu, W.; Zhang, B.; Wang, B.; Dong, K.; Liu, B.; Sun, J.; Ren, T.; Li, Z.; Yang, H.; et al. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525* **2024**.
60. Infosecurity Magazine. DeepSeek-R1 and AI Security: How DeepSeek Ensures Safe AI Deployment. https://www.infosecurity-magazine.com/news/deepseek-r1-security/, 2025.
61. Alibaba Cloud. Qwen: Generative AI Model by Alibaba Cloud. https://www.alibabacloud.com/en/solutions/generative-ai/qwen, 2024.