ISSN: 3067-8307
Review Article
World Journal of Applied Mathematics and Statistics

Tracing the Evolution of Artificial Intelligence: A Review of Tools, Frameworks, and Technologies (1950–2025)

Gurpreet Singh[1]*, Trina Banerjee[2] and Nishaa[3]

[1]*Woosong University, Republic of Korea*
[2]*Saks Global, India*
[3]*Independent Researcher, India*

***Corresponding author:** Gurpreet Singh, Woosong University, Republic of Korea.

Submitted: 11 November 2025 **Accepted:** 18 November 2025 **Published:** 26 November 2025

https://doi.org/10.63620/MKWJAMS.2025.1023

***Citation:** Singh, G., Banerjee, T., & Nishaa. (2025). Tracing the Evolution of Artificial Intelligence: A Review of Tools, Frameworks, and Technologies (1950–2025). Wor Jour of Appl Math and Sta, 1(4), 01-28.*

Abstract
Artificial Intelligence (AI) has evolved remarkably over the past seven decades, transforming from simple rule-based systems into complex multimodal and generative frameworks capable of reasoning, creativity, and perception. This review traces the chronological development of AI tools, highlighting key milestones that shaped the field—from the early symbolic programs like Logic Theorist and ELIZA to the emergence of modern large-scale models such as GPT-4, Gemini, and Claude. The study explores the progression across distinct eras: the foundational period of symbolic reasoning (1940s–1970s), the rise of machine learning and statistical modeling (1980s–2000s), the deep learning revolution (2010s), and the recent explosion of generative and multimodal systems (2020–2025). Each phase reflects a major shift in how intelligence is defined, represented, and implemented—from handcrafted logic to data-driven learning and now to context-aware multimodal understanding. By reviewing over fifty significant AI tools and frameworks, this paper provides a comprehensive overview of how incremental innovations in computation, data availability, and model architecture have collectively enabled the current state of AI. The work concludes with insights on how this evolution paves the way for the next generation of agentic and real-time AI systems capable of seamless interaction across text, image, audio, and video modalities.

Keywords: Artificial Intelligence; Deep Learning; Generative Models; Multimodal Systems; Ai Evolution.

Introduction

Artificial Intelligence (AI) has become one of the most transformative forces of the 21st century, shaping industries, research, and everyday life in ways that were once thought impossible. From simple logic-based programs created in university labs to today's generative systems capable of producing human-like text, images, and even videos, the journey of AI represents not just technological progress but also humanity's evolving understanding of intelligence itself. Each era of AI—from symbolic reasoning and rule-based systems to neural networks, deep learning, and now multimodal generative models—has introduced new ways of learning, interacting, and problem-solving.

The purpose of this paper is to provide a comprehensive and chronological review of the evolution of AI tools and frameworks, capturing how each innovation built upon the previous generation to form the foundation of modern artificial intelligence. While many studies focus on specific branches of AI such as computer vision, natural language processing, or robotics, few have presented a complete timeline connecting the earliest symbolic systems to the current multimodal and agentic era. This review aims to fill that gap by examining over seven decades of AI development—from early expert systems like MYCIN and DENDRAL to advanced generative models like GPT-4, Gemini, and Sora.

Writing this paper in 2025 holds particular significance. The world is now witnessing a rapid convergence of text, image, speech, and video understanding through multimodal AI models, making it essential to reflect on how we arrived at this point and what lessons past innovations can teach us. The pace of AI progress often makes older technologies seem obsolete, yet many of today's breakthroughs are deeply rooted in the principles and experiments of earlier decades. Understanding this historical evolution helps researchers, educators, and practitioners appreciate the foundations of current AI tools and foresee the directions future technologies might take.

Ultimately, this paper serves as both a historical reflection and an educational guide, offering readers a unified perspective on how AI tools have matured from isolated programs into interconnected, intelligent ecosystems. By mapping this evolution, the study underscores the continuity of innovation that defines artificial intelligence—not as a sudden revolution, but as a long, collaborative journey of human curiosity and technological advancement.

Early Foundations (1940s–1970s)

Logic Theorist (1956)

The Logic Theorist, developed in 1956 by Allen Newell, Herbert A. Simon, and Cliff Shaw at the RAND Corporation, is widely recognized as the first artificial intelligence (AI) program purposely designed to mimic human problem-solving and reasoning skills. Its primary goal was to prove mathematical theorems, specifically those found in the seminal work Principia Mathematica by Alfred North Whitehead and Bertrand Russell. The program demonstrated that a machine could engage in automated reasoning by successfully proving 38 of the first 52 theorems in chapter two of Principia Mathematica. Remarkably, the Logic Theorist not only replicated human logic but also discovered new, sometimes shorter, proofs for some theorems, exemplifying the ability of AI to surpass human intellectual processes in specific domains [1].

The development of the Logic Theorist was grounded in symbolic logic and heuristic search—a method that applies rules of thumb to explore possible solutions in a vast search space. The program represented logical expressions symbolically and searched through combinations of these expressions to construct proofs. It used heuristics to efficiently navigate the tree of possible inferences, thus avoiding an exhaustive and computationally prohibitive search through all branches. This heuristic approach was inspired by how humans tackle complex problems, prioritizing promising paths based on certain criteria rather than brute-force search [2].

Architecturally, the Logic Theorist consisted primarily of two components: a knowledge base containing axioms and previously proven theorems, and an inference engine that applied logical rules to generate new theorems. The proof search was conducted by generating sub-proofs for propositions through symbolic manipulations. Although programming languages and computational resources were very primitive at the time, the team implemented the Logic Theorist using handwritten cards that were distributed among collaborators and later run on computers available at RAND [3]. This development also led to the creation of the Information Processing Language (IPL), which facilitated list processing and symbolic computation—precursors to programming languages such as LISP, which became fundamental in AI research [4].

Mathematically, the Logic Theorist operated within propositional calculus as formulated in Principia Mathematica. Its proofs often utilized methods such as proof by contradiction, where the algorithm would assume the negation of a theorem and derive contradictions to verify its truth. By starting from axioms and applying inference rules, the program mechanized the deductive reasoning process typical of mathematical proof. The discovery of novel proofs for certain theorems, such as theorem 2.85, highlighted its capability not only to replicate but to improve upon human mathematical reasoning [5].

The impact of the Logic Theorist on AI and cognitive science was profound. It marked the first practical demonstration that machines could perform high-order intellectual tasks involving symbolic reasoning. It laid foundational concepts such as heuristic search, symbolic processing, and the notion that reasoning could be mechanized, influencing subsequent AI programs and research. Herbert Simon and Allen Newell, the principal developers, were later recognized with Turing Awards for their pioneering contributions to AI and computer science. The Logic Theorist remains a landmark in AI history, illustrating the power of symbolic AI and heuristic-driven problem solving in the early quest to imitate human thought through machines

Eliza (1966)

ELIZA, developed in 1966 by Joseph Weizenbaum at MIT, is one of the earliest and most influential natural language processing programs and is widely regarded as the first chatbot to simulate human conversation [6]. Unlike later conversational agents built on complex machine learning models, ELIZA operated fundamentally through pattern matching and substitution rules, using scripts designed to give the illusion of understanding without true comprehension. The most famous script, known as "DOCTOR," simulated a Rogerian psychotherapist by reflecting users' statements back to them in the form of questions, encouraging users to continue dialogue . For instance, when a user expressed feelings of sadness or worry, ELIZA would respond with inquiries prompting further reflection, employing simple yet effective mechanisms to seem empathetic and human-like.

ELIZA's architecture was based on symbolic processing where user inputs were analyzed for keywords or phrases, and then matching scripted patterns were triggered to generate predefined responses. It did not involve semantic or contextual understanding, which distinguished it from genuine human communication. Nevertheless, the program's simplistic style led to what became known as the "ELIZA effect," [7] where users attributed human-like understanding and emotions to the program despite its mechanical nature. This phenomenon highlighted important psychological and philosophical questions about human interaction with machines and the nature of intelligence. Interestingly, Weizenbaum himself was surprised and troubled by how readily some users, including his own secretary, formed emotional connections with ELIZA, underscoring the powerful illusion of understanding it created.

The development of ELIZA marked a significant milestone in artificial intelligence and human- computer interaction because it demonstrated that computers could engage users in seemingly mean- ingful conversation through language-based interfaces. Its creation also helped lay the groundwork for research in natural language understanding, dialogue systems, and chatbot development. ELIZA's influence extended beyond technological innovation; it sparked critical debates about machine intel- ligence, ethics, and the limitations of AI, since the program could simulate empathy without actual cognitive or emotional processing. Weizenbaum stressed that ELIZA was a tool, not a sentient entity, cautioning against overestimating what machines can replicate in terms of human thought and feeling [8].

In summary, ELIZA represents a pioneering experiment in AI programming that combined early symbolic processing techniques and heuristic scripting to simulate conversation. While mathematically it did not engage in logical deduction, it excelled in procedural language pattern matching, which was enough to make a lasting impact on AI, natural language processing, and cognitive science. Its legacy continues in modern chatbots that have since evolved with more sophisticated models, but ELIZA remains a foundational artifact demonstrating the potentials and boundaries of early conversational AI [9]

Shrdlu (1970)
SHRDLU, developed between 1968 and 1970 by Terry Winograd at MIT as part of his PhD thesis, represents a pioneering natural language understanding system that could interpret and execute commands within a simplified virtual environment known as the "blocks world." This virtual world consisted of various geometric shapes—blocks, pyramids, and boxes of different colors—that SHRDLU manipulated in response to user instructions phrased in natural English. The program could perform tasks such as moving objects, stacking blocks, and answering questions about the state of the environment, such as "What is on the table?" or "Is there a pyramid on the block?" The user interacted with SHRDLU through a dialogue interface that allowed both commands and queries, making it an early example of an interactive AI system capable of understanding and reasoning about language and its referents in a controlled world [10].

From a technical standpoint, SHRDLU combined syntax parsing, semantic interpretation, and a form of procedural knowledge representation to understand instructions and generate responses. It was implemented mainly in Lisp and Micro Planner on a DEC PDP-6 computer. The system parsed English commands, mapped them onto internal logical representations of objects and their properties in the blocks world [11] , and then used rules to manipulate these objects or generate language responses. This architecture showcased an integrated approach to natural language processing, combining syntax, semantics, and pragmatics within a finite domain. SHRDLU's knowledge base was dynamic, maintaining an up-to-date model of the world's state, which it could modify as it executed actions or learned new commands. The logical foundation often related to first-order logic, where objects, properties, and relations were explicitly represented, enabling reasoning over the set of facts.

Beyond its technical achievements, SHRDLU critically demonstrated the challenges and potential of language understanding in AI. Winograd's work exposed the brittleness of symbolic systems when faced with the complexity and ambiguity of natural language outside a restricted domain. Yet, SHRDLU's ability to follow complex instructions and engage in clarifying dialogues to resolve ambiguities was groundbreaking, highlighting the possibility of interactive AI systems that understand human language contextually. It influenced subsequent research in computational linguistics, human- computer interaction, and AI, laying a foundation for modern dialogue systems, virtual assistants, and robotic control by natural language. SHRDLU's legacy extends into contemporary natural language processing by inspiring approaches to syntactic and semantic integration and knowledge representation in AI, demonstrating how procedural representations can bridge language and action within a defined world framework [12].

This comprehensive understanding of SHRDLU's development, architecture, functionality, and impact should serve well for your research paper, situating it as a landmark system in natural language understanding and AI history.

Dendral (1965–1970)
Developed between 1965 and 1970 at Stanford University by Edward Feigenbaum, Bruce Buchanan, Joshua Lederberg, and Carl Djerassi, DENDRAL is recognized as the first successful expert system and a landmark in artificial intelligence history. Designed primarily as a chemical analy- sis tool, DENDRAL aimed to assist organic chemists in identifying unknown molecular structures by analyzing mass spectrometry data. Unlike prior AI systems, DENDRAL encoded expert knowledge in chemistry as heuristic rules and used this domain-specific expertise to automate the process of hypothesis generation and evaluation—effectively emulating the decision-making process of skilled human chemists. This enabled the system not only to generate possible molecular structures but also to predict their corresponding mass spectra and compare them with the experimental data to select the most plausible hypotheses [13].

The architecture of DENDRAL was characterized by a clear separation between its knowledge base and inference engine, forming a paradigm later foundational to expert systems. Its knowledge base contained encoded chemical heuristics and domain rules, while the inference engine orchestrated the generation and evaluation of candidate molecular structures through a heuristic search process. The system consisted mainly of two programs: Heuristic Dendral, which performed structure elucidation using domain-specific heuristics, and Meta-Dendral, which learned new rules by analyzing patterns in chemical data, pioneering early forms of machine learning. Technically, DENDRAL combined symbolic reasoning with heuristic search strategies to prune the vast search space of chemical structures, applying constraints and domain knowledge to improve computational efficiency and accuracy [14]. Mathematically, DENDRAL relied on combinatorial and graph-theoretic methods to represent molecular structures as cyclic graphs and trees, aligning with its etymological roots ("dendron" meaning tree in Greek). It employed algorithms that enumerated possible configurations constrained by chemical valence rules and mass spectral data, effectively narrowing down the candidates through heuristic pruning. By translating chemical knowledge into formal rules and representing candidate molecules structurally, DENDRAL bridged symbolic AI with practical chemical problem-solving.

The significance of DENDRAL extends beyond its application, as it pioneered the concept and successful implementation of expert systems—computer programs that encapsulate expert-level knowledge and reasoning ability within a specific domain. It demonstrated AI's potential to enhance scientific discovery and decision-making by explicitly codifying expert heuristics into programmable knowledge bases. Furthermore, DENDRAL influenced subsequent AI developments, including MYCIN and other domain-specific expert systems, and helped establish research paradigms in AI knowledge engineering and human-computer interaction. Its success bolstered funding and interest in AI during the 1970s and fundamentally shaped the trajectory of knowledge-based systems in computer science and expert decision support [15].

This detailed overview captures DENDRAL's impact as a pioneering expert system in chemical analysis through heuristic symbolic reasoning, foundational to AI's evolution in expert knowledge representation and automated problem-solving.

Mycin (1972)
MYCIN, developed in the early 1970s at Stanford University by Edward Shortliffe and colleagues, is a pioneering rule-based expert system designed to assist physicians in diagnosing bacterial infec- tions and recommending appropriate antibiotic treatments. Originating from the Stanford Heuristic Programming Project, MYCIN employed a knowledge base of approximately 600 production rules encoding the clinical decision-making expertise of infectious disease specialists. The system operated through a backward chaining inference engine that interactively queried physicians with a series of simple yes/no or text-based questions about the patient's symptoms and lab results. Based on the gathered data, MYCIN produced a ranked list of potential causative bacteria with associated confidence levels and provided tailored antibiotic therapy suggestions, adjusting dosages based on patient-specific factors such as body weight [16].

Technically, MYCIN's architecture distinguished itself by clearly separating its inference proce- dures from its domain-specific knowledge, embodying a modular approach that became foundational for later expert systems. The system implemented a certainty factor model to handle uncertainty in medical diagnosis, representing the confidence in rules and conclusions, although this model was heuristic rather than strictly Bayesian due to computational and practical constraints. MYCIN was also notable for its explanatory capabilities, where it could justify its recommendations by tracing back through the rules and questions that led to its conclusions, addressing concerns about transparency and trust in AI systems for medical decision-making [17].

From a mathematical and logical perspective, MYCIN relied on production rules—if-then state- ments representing clinical knowledge—and heuristic-driven backward chaining to efficiently search the space of diagnostic possibilities. The certainty factor calculus combined evidential strengths from multiple rules to arrive at probabilistic-like confidence measures in diagnoses and recommenda- tions. While these certainty factors lacked rigorous statistical foundations, they provided a practical framework for handling medical diagnostic uncertainty.

MYCIN's impact on AI and medicine was profound, demonstrating that expert knowledge in specialized domains could be formalized and leveraged by computer systems to perform tasks at the level of human specialists. Its success inspired numerous subsequent expert systems and contributed significantly to the development of knowledge engineering, rule-based reasoning, and explanation facilities in AI. MYCIN also sparked discussions on the ethical and practical implications of AI in clinical settings, particularly regarding the role of decision support versus autonomous diagnosis. Although constrained by the computational power and data availability of its time, MYCIN laid the groundwork for modern clinical decision support systems and remains a seminal example of early AI applied to real-world expertise [18]. This comprehensive account captures MYCIN's development, architecture, operation, mathemati- cal underpinnings, and its lasting significance as a foundation for rule-based AI systems, making it a critical milestone in the history of artificial intelligence and medical informatics.

Prolog (1972)
Prolog, created in 1972 by Alain Colmerauer and Philippe Roussel at Aix-Marseille University in France, is a logic programming language foundational to artificial intelligence and natural language processing tasks. The name Prolog derives from the French phrase "Programmation en Logique," meaning programming in logic. Rooted in first-order predicate logic, Prolog allows programmers to express knowledge declaratively, using facts and rules rather than explicit procedural code. The language's execution is driven by a goal-directed search process implementing resolution theorem proving with Horn clauses, which it processes through a mechanism called backward chaining. This enables Prolog to perform symbolic reasoning, automated theorem proving, and natural language understanding, making it particularly suited for AI applications [19].

The language emerged from combining two efforts: Alain Colmerauer's focus on natural language processing and Robert Kowalski's theoretical work on the procedural interpretation of logic. The original Prolog system included an interpreter written in Algol-W and later versions influenced by David H.D. Warren, who developed the Warren Abstract Machine (WAM), a standard virtual machine architecture for efficient Prolog implementation. This contributed significantly to Prolog's widespread adoption and the establishment of the Edinburgh syntax standard that most Prolog implementations follow. Prolog's logical foundation centers on Horn clauses that define relations, with computations being queries that the interpreter attempts to satisfy by proving them from the known facts and rules [20].

Prolog's computational model is distinguished by features like unification, backtracking, and negation as failure, which together allow it to explore the space of potential solutions effectively. Mathematically, Prolog programs can be viewed as logic formulas in clausal form and the resolution proof method underpins its operation. The declarative style facilitates expressing complex information and constraints naturally, enabling applications in expert systems, theorem proving, language parsing, knowledge representation, and automated planning.

Historically, Prolog gained international prominence during the 1980s particularly with the Japanese Fifth Generation Computer Systems project, which sought to leverage logic programming for advanced AI on parallel computers. While Prolog faced competition and technical challenges, its conceptual clarity and powerful expressiveness secured its lasting place in AI programming. Today, Prolog remains a prominent symbolic programming language, well-suited to tasks demanding sophisticated pattern matching, reasoning, and rule-based logic processing, continuing to underpin research and applications in artificial intelligence and computational linguistics [21].

Lisp (1958)
LISP, created in 1958 by John McCarthy at MIT, is one of the oldest and most influential pro- gramming languages in the field of artificial intelligence. McCarthy designed LISP with the explicit goal of providing a powerful and flexible language specifically suited for AI research, particularly symbolic manipulation and processing of list structures. The language introduced several grounds- breaking concepts, such as recursive functions, dynamic typing, and automatic memory management through garbage collection. Its primary data structure, the list, and its associated operations—car and cdr—allowed natural and efficient expression of symbolic computation, which is central to AI [22].

Initially, McCarthy proposed a notation called "M-expressions" to make LISP more readable, but it was quickly abandoned in favor of "S-expressions," a more straightforward parenthesized prefix notation that became the hallmark of LISP's syntax. The first LISP interpreter was implemented by Steve Russell on an IBM 704, surprising McCarthy by demonstrating that the eval function—a core evaluation procedure—could be directly compiled into machine code, enabling practical execution. LISP's design was mathematically rooted in the theory of recursive functions and lambda calculus, which provided a formal foundation for defining computable functions and reasoning about process execution [23].

Over the decades, LISP became the dominant AI programming language because of its unpar- alleled ability to handle symbolic information, its extensibility through macros, and its interactive development environment. It powered pioneering AI projects such as SHRDLU and expert systems, and influenced other languages like Scheme and Common Lisp. LISP introduced concepts critical to AI, including symbolic reasoning, code as data (homoiconicity), and advanced macro systems, which helped AI researchers experiment with new ideas efficiently. Common Lisp, standardized in the 1980s and 1990s, unified various dialects and remains widely used in AI, research, and industry.

LISP's mathematical essence lies in its recursive function definitions, symbolic lists, and evalua- tion semantics, which collectively enabled it to express complex AI algorithms elegantly. Its legacy continues to permeate AI research, functional programming, and symbolic computation, making it a foundational technology in the history and development of artificial intelligence [24].

This overview encapsulates LISP's historical context, innovative design principles, mathematical underpinnings, and its critical role as the dominant AI programming language for decades.

The Rise of Machine Learning (1980s–2000s)

Soar (1983)
SOAR, developed in 1983 by John Laird, Allen Newell, and Paul Rosenbloom at Carnegie Mellon University, is a pioneering cognitive architecture designed to model human-like reasoning and general intelligence. Rooted deeply in cognitive science and artificial intelligence, SOAR aims to create fixed computational building blocks for general intelligent agents capable of performing a wide array of tasks such as decision-making, problem-solving, planning, and natural language understanding. The architecture embodies a unified theory of cognition, evolving from Allen Newell's Problem Space Hypothesis—a foundational AI theory stating that all goal-directed behavior can be framed as search within a space of possible states. SOAR operationalizes this by continuously selecting and applying operators to change an agent's state, analogous to how humans approach problems step-by-step [25]. SOAR's architecture integrates procedural memory (knowledge of how to do things) with working memory (representation of the current situation), enabling a dynamic cognitive cycle. Proce- dural knowledge is encoded as if-then production rules (condition-action pairs) that match against the contents of working memory. Unlike other systems, SOAR fires all matching rules in parallel, allowing concurrent context-dependent retrieval of knowledge. When the system encounters an impasse—lacking knowledge to proceed—SOAR automatically creates a substate that recursively applies the same problem-solving process, leading to the generation of subgoals and hierarchical task decomposition. This universal subgoaling approach naturally models complex cognitive behaviors including planning and learning. The working memory itself is structured as a symbolic graph rooted in the current state, facilitating flexible representation of knowledge [26].

Mathematically, SOAR's computation is grounded in state-transition systems, production rule matching, and search algorithms navigating complex problem spaces. The production system supports parallel rule matching while ensuring that behavior is decomposed into primitive operator applications approximating human reaction times (5˜0ms per step). The substate recursion forms a sophisticated mathematical structure supporting hierarchical problem-solving, setting SOAR apart from prior architectures. Its design reflects the interplay of symbolic AI with psychological realism, striving to unify AI capabilities and cognitive modeling [27].

Over the decades, SOAR has evolved into a comprehensive cognitive architecture widely em- ployed in AI research to build intelligent agents and model human behavior in a range of applications—from robotics to gaming to natural language processing. Maintained today by John Laird's research group at the University of Michigan, SOAR remains a seminal reference point for cognitive architectures, continuing to influence theories of general intelligence by bridging formal computational models with empirical cognitive science insights. This detailed exploration covers SOAR's theoretical foundations, developmental history, architec- ture, mathematical underpinnings, and its significance in AI and cognitive science [28].

Alvinn (1989)
ALVINN (Autonomous Land Vehicle in a Neural Network), developed in 1989 at Carnegie Mellon University by Dean Pomerleau, is a landmark early neural network system designed for autonomous vehicle navigation and road following. As one of the first practical applications of artificial neural networks to self-driving cars, ALVINN was built around a three-layer backpropagation network that took as input a combination of senso-

ry data from a forward-facing camera and a laser range finder, processing images and depth information to produce steering commands for the vehicle. The architecture comprised roughly 1217 input units, a hidden layer with 29 units, and an output layer with 46 units dedicated to representing a range of possible steering angles, along with feedback units to integrate temporal context [29].

Trained initially on simulated road images, ALVINN used supervised learning with backprop- agation to optimize its weights, enabling it to predict the required steering angle to follow the road accurately. The training leveraged data augmentation techniques, generating shifted versions of images to robustly handle diverse driving scenarios and prevent the system from failing when deviating from the road centerline. Operationally, ALVINN was deployed on the NAVLAB autonomous test vehicle, demonstrating its ability to navigate complex outdoor environments—including varied weather condi- tions—at speeds competitive with traditional computer vision approaches of the time. This success illustrated the promise of adaptive, data-driven models in autonomous navigation, surpassing rigid rule-based systems by learning representations tailored dynamically to sensory inputs.

Mathematically [30], ALVINN employed a feedforward neural network with supervised gradient descent optimization (backpropagation) to minimize prediction errors between the network output and actual steering commands. The network model embodied non-linear function approximation, capturing the complex mapping from high-dimensional sensory input spaces (images and range data) to control actions. The use of feedback units introduced an element of temporal memory, allowing the network to incorporate information about the recent past into its decision-making process—related conceptually to what would later be developed as recurrent neural architectures.

ALVINN's introduction was a key milestone in AI and robotics, illustrating how neural networks could be harnessed for real-world control problems under uncertainty and noise. It heralded the feasibility of end-to-end learning for vehicle control, influencing future developments in autonomous driving technology and machine learning-driven robotics. The system highlighted the importance of sensor fusion, adaptive learning, and robust training strategies in autonomous navigation, and its design principles form the basis for many modern AI-based self-driving systems [31].

Backpropagation Algorithm (1986)
The backpropagation algorithm, popularized in 1986 by David Rumelhart, Geoffrey Hinton, and Ronald Williams, revolutionized the training of artificial neural networks by providing an efficient method to compute gradients of the loss function with respect to weights and biases for multilayered networks. Prior to this breakthrough, training deep networks was computationally infeasible due to the difficulty in attributing errors to internal layers. Backpropagation operationalizes gradient descent in a highly scalable way, allowing error signals from the output layer to be propagated backward through the network layers by systematically applying the chain rule of calculus [32].

The algorithm works in two main phases during each training iteration. In the forward pass, the network computes the output by propagating inputs through successive layers via weighted connections and activation functions. In the backward pass, the algorithm calculates the gradient of the loss function relative to each weight by propagating the error backward from the output layer to the input layer. These gradients are then used to update network parameters in the direction that minimizes the output error. Mathematically, for a weight wjkl connecting neuron k in layer l − 1 to neuron j in layer l, the update step corresponds to computing the partial derivative of the cost C with respect to that weight, $\frac{\partial C}{\partial w_{jkl}}$, and using it in gradient descent:

$$w_{jkl} \leftarrow w_{jkl} - \eta \frac{\partial C}{\partial w_{jkl}}$$

where η is the learning rate. This process is repeated iteratively over examples, applying the chain rule to efficiently allocate error contributions among all weights.

Backpropagation's significance lies not only in its algorithmic efficiency but also in its enabling of modern deep learning. It permits neural networks to learn internal representations automatically, making them capable of solving highly complex and nonlinear tasks including image recognition, speech processing, and natural language understanding. The 1986 paper by Rumelhart, Hinton, and Williams marked a turning point in AI because it provided a practical and scalable learning algorithm that forms the foundation of nearly all contemporary deep learning architectures. Since then, backpropagation has become the workhorse of neural network training, supported by abundant optimizations, regularization methods, and variants that have further expanded the reach of AI solutions [33].

In essence, backpropagation combines deep mathematical concepts from calculus with computa- tional strategies to effectively optimize multilayer neural networks, ushering in the era of powerful, adaptive artificial intelligence systems.

Weka (1993)
WEKA (Waikato Environment for Knowledge Analysis), developed starting in 1993 at the Uni- versity of Waikato in New Zealand, is a comprehensive, open-source machine learning and data mining software suite widely used for teaching, research, and practical data analysis. Created by a team led by Professor Ian H. Witten and colleagues, WEKA was born from a desire to provide an accessible platform that integrates a large collection of machine learning algorithms with tools for data preprocessing, visualization, and evaluation, all accessible through user-friendly graphical interfaces. Originally, WEKA was developed with agricultural and horticultural data applications in mind, but it has since evolved to support a diverse array of domains and data types [34].

WEKA's architecture centers around modular components that perform key data mining tasks: classification, regression, clustering, association rule mining, and feature selection. It supports exten- sive data preprocessing options such as cleaning, normalization, and attribute transformation within a flexible pipeline, enhancing model performance on real-world data. One of WEKA's hallmark features is its graphical user interface, which includes the Explorer, Experimenter, and KnowledgeFlow, en- abling users to design, test, and visualize machine learning workflows interactively without requiring deep programming expertise. For advanced users, WEKA provides an API and scripting capabilities to integrate its algorithms into custom applications [35].

Technically, WEKA is implemented in Java, making it highly portable across operating systems and easy to extend with new algorithms. It processes input in the attribute-relation file format (ARFF) but also supports various data formats through database connectivity and file import/export. WEKA's comprehensive collection of algorithms includes decision trees, support vector machines, neural networks, Bayesian classifiers, clustering methods like k-means, and association rule algorithms, making it a versatile toolset for empirical machine learning experimentation and deployment.

Mathematically, WEKA's algorithms collectively represent a spectrum of machine learning meth- ods, from statistical models and decision theory to heuristic and optimization-based techniques. The suite supports comparative evaluation approaches including cross-validation and receiver operating characteristic (ROC) analysis, facilitating rigorous empirical assessment of model performance—an essential pillar of machine learning research methodology.

WEKA's impact is significant as it democratized access to machine learning techniques by lowering the barrier to entry and standardizing experimentation workflows. It remains a cornerstone in machine learning education and research, continuously updated by the University of Waikato community and an active global user base. Its open-source nature has stimulated wide adoption in academia and industry, fostering innovation, collaboration, and reproducibility in the machine learning field [36].

Support Vector Machines (SVM, 1995)
Support Vector Machines (SVM), a transformative method in machine learning, were introduced in their widely recognized form through the seminal 1995 paper by Corinna Cortes and Vladimir Vapnik at AT&T Bell Laboratories. Building on foundational work by Vapnik and Alexey Chervonenkis dating back to the 1960s and 1970s, SVM formalized a powerful approach to binary classification by finding the optimal separating hyperplane that maximizes the margin—the distance between the hyperplane and the nearest data points of each class, called support vectors. This concept of maximum-margin classification provides robust generalization on unseen data, improving predictive accuracy and reducing overfitting compared to many earlier algorithms [37].

Mathematically, given training data points $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ where $\mathbf{x}_i \in \mathbb{R}^p$ and labels $y_i \in \{-1, +1\}$,
SVM finds a hyperplane defined by $w \cdot x + b = 0$ such that the margin $\frac{2}{\|\mathbf{w}\|}$ is maximized under the constraints

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad \forall i.$$

The resulting convex optimization problem can be solved efficiently using quadratic programming. To handle non-linearly separable data, Cortes and Vapnik introduced the "soft margin" formulation, allowing some misclassification controlled by slack variables, balancing margin maximization and error minimization. Furthermore, the kernel trick—developed in continuation by Boser, Guyon, and Vapnik in 1992—enables SVMs to operate implicitly in high-dimensional feature spaces using kernel functions like polynomial or radial basis functions, thereby extending SVMs to complex, nonlinear classification tasks.

SVM's rigorous roots in statistical learning theory give it strong theoretical guarantees on gen- eralization and consistency, distinguishing it from heuristic-based classification methods of its time. Its algorithmic efficiency, versatility across domains, and theoretical elegance made it a core machine learning tool from the mid-1990s onward, with applications spanning image recognition, bioinfor- matics, text classification, and many others. The 1995 paper by Cortes and Vapnik is regarded as a breakthrough that not only provided a practically viable classification algorithm but also solidified the importance of large-margin classifiers in supervised learning.

Today, SVM remains a backbone method in classical machine learning, often contrasted with modern deep learning approaches but still valued for its interpretability, mathematical clarity, and performance on smaller datasets. Its influence continues in enhancing kernel methods, support vector regression, and structured prediction tasks, making it a foundational pillar in the evolution of AI and machine learning [38]

OpenCV (2000)
OpenCV (Open-Source Computer Vision Library) is an influential open-source library for com- puter vision and image processing that originated as an Intel Research initiative in 1999 under the leadership of Gary Bradski. Its first public release occurred in 2000, aiming from the start to democ- ratize computer vision by providing optimized, portable, and accessible software infrastructure for a broad array of vision tasks. This initiative sought to eliminate duplication of effort by offering a comprehensive and efficient collection of algorithms for real-time image and video analysis, benefiting researchers, developers, and commercial applications alike [39].

Developed primarily in C and C++ with bindings for Python, Java, and other languages, OpenCV has evolved into a cross-platform [40] library widely used in academia and industry for its breadth and efficiency. Its architecture includes modular components for image processing, feature detection, object recognition, camera calibration, machine learning, and later, integration of deep learning modules for neural network inference. OpenCV's development ethos stresses performance optimization, with ongoing enhancements including multi-core processing, GPU acceleration, and support for new hardware platforms to meet the demands of modern applications such as autonomous vehicles, augmented reality, robotics, and medical imaging.

Mathematically, OpenCV encapsulates a wide array of methods from classical computer vision like edge detection, geometric transformations, and stereo vision to statistical and machine learning techniques, including clustering, support vector machines, and neural networks. It provides imple- mentation for linear algebra, matrix operations, filtering, and advanced models used in image analysis, facilitating both foundational research and applied system development.

Over the years, OpenCV's stewardship transitioned from Intel to Willow Garage and Itseez before returning under Intel's portfolio via acquisition, with an active community and the non- profit OpenCV.org foundation now guiding its continual evolution. Its broad adoption, sustained development, and extensible architecture make OpenCV a cornerstone tool that has significantly influenced the growth and accessibility of computer vision technologies for more than two decades [41].

Matlab Ai Toolbox (2000s)
MATLAB, originally developed in the late 1970s by Cleve Moler and commercialized through MathWorks from the mid-1980s onwards, is a powerful numeric computing environment and pro- gramming language widely adopted in engineering, science, and applied mathematics. The MATLAB AI Toolbox, introduced progressively throughout the 2000s, significantly expanded MATLAB's capa- bilities by providing a dedicated platform for developing and experimenting with machine learning, artificial intelligence, and control algorithms. These toolboxes integrated a rich variety of functions and apps supporting classification, regression, clustering, neural networks, reinforcement learning, and deep learning, along with tools for data preparation, visualization, algorithm tuning, and deployment [42].

Architecturally, the AI Toolbox leverages MATLAB's matrix-based language to allow users to express ML and AI algorithms succinctly, combining high-level programming with efficient built-in computational routines. The toolbox supports both algorithmic prototyping and production-grade code generation, enabling seamless transitions from research experiments to implementation. MAT- LAB's interactive environment with integrated plotting and coding facilitates fast iteration, debugging, and data exploration—making it particularly favored in academic research and industrial applica- tions focused on control systems, autonomous vehicles, robotics, signal processing, and biomedical engineering.

Mathematically, MATLAB AI Toolbox algorithms encompass classical statistical models, opti- mization routines, neural network training via backpropagation [43], and advanced deep learning architectures. The platform provides robust support for linear and nonlinear system modeling, state estimation, and adaptive control theory, underpinned by extensive numerical libraries for matrix decomposition, eigenvalue computation, numerical integration, and optimization algorithms such as gradient descent, conjugate gradient, and quasi-Newton methods. Deep learning support inte- grates seamlessly with popular frameworks like TensorFlow and PyTorch, allowing MATLAB users to leverage and customize cutting-edge network architectures.

The MATLAB AI Toolbox has become a cornerstone in the AI research and development ecosystem due to its comprehensive functionality, ease of use, and powerful computational engine. It promotes reproducibility and collaboration across disciplines by combining rich algorithmic libraries with user- friendly interactive features. Continuously enhanced by MathWorks, the toolbox plays a critical role in advancing AI technologies, enabling engineers and scientists to develop new AI methods and apply them to complex real-world problems efficiently [44] .

Scikit-Learn (2007–2010)
Scikit-learn (also known as sklearn), developed between 2007 and 2010, is a widely acclaimed open-source machine learning library for the Python programming language designed to provide accessible and efficient tools for data mining and data analysis. The project began as a Google Summer of Code initiative by French data scientist David Cournapeau in 2007, originally intended as a scientific toolkit extension to the SciPy ecosystem. In 2010, key contributors from the French Institute for Research in Computer Science and Automation (INRIA)—Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, and Vincent Michel—took over the development, leading to the release of the first public version on February 1, 2010.

Scikit-learn's architecture is built around a consistent, user-friendly API that integrates a broad selection of supervised and unsupervised learning algorithms, including classification, regression, clustering, dimensionality reduction, and model selection. It builds on the foundational scientific com- puting libraries in Python, notably NumPy and SciPy, enabling efficient numerical computation and making it a natural choice for researchers and data scientists. The library also provides comprehensive documentation, concrete examples, and seamless interoperability with other Python data tools such as pandas and Matplotlib.

Mathematically, scikit-learn encompasses a diverse range of algorithms spanning linear models (like linear and logistic regression), support vector machines, ensemble methods (random forests, gradient boosting), clustering methods (k-means, DBSCAN), and manifold learning, among others. It supports parameter tuning, model validation through cross-validation, and evaluation metrics, facilitating rigorous experimental workflows in machine learning research and industry applications. Scikit-learn's impact has been significant in democratizing machine learning, making state-of-the-art algorithms easy to experiment with and deploy while fostering a large active community that continuously contributes improvements. Its design philosophy emphasizes simplicity, efficiency, and versatility, contributing to its widespread adoption across academic and commercial sectors, advancing both education and practical AI solutions worldwide.

Deep Learning & Neural Network Revolution (2010s)
Theano (2010)
Theano is an open-source Python library developed at the Montreal Institute for Learning Al- gorithms (MILA) at the Université de Montréal, first released publicly in 2007. It was designed as a powerful numerical computation framework that specializes in defining, optimizing, and efficiently evaluating mathematical expressions, particularly those involving multi-dimensional arrays, which are common in machine learning and deep learning models. Its core innovation lay in symbolic computation, where users define computational graphs symbolically, allowing Theano to apply so- phisticated optimizations and compile efficient code targeted for CPU or GPU hardware, crucially enabling GPU-accelerated training of complex neural networks [45].

Developed under the leadership of renowned AI researcher Yoshua Bengio and his team, Theano introduced automated differentiation capabilities, simplifying the process of computing gradients necessary for training deep learning models using backpropagation. By abstracting the mathematical details and transparently leveraging hardware acceleration, Theano helped researchers explore and implement novel neural architectures and advanced learning algorithms with greater ease and speed than was previously possible.

Mathematically, Theano's strength comes from representing computations as directed acyclic graphs with nodes representing mathematical operations and edges representing data dependencies, facilitating efficient symbolic differentiation through the chain rule and optimization of the com- putation graph before execution. This design allowed it to seamlessly integrate with the NumPy ecosystem while offering dynamic C code generation and extensive unit testing for numerical stability and reliability.

Though major development ceased in 2017 due to emerging competitors like TensorFlow and PyTorch, Theano's influence endures—it served as the computational backend for popular deep learning libraries such as Keras and Lasagne, and its pioneering work in symbolic graph optimizations and automatic differentiation continue to underpin modern deep learning frameworks. The open-source community and projects like PyTensor (a fork and continuation) have maintained its legacy, ensuring its foundational ideas persist in advancing AI research and applications.

Overall, Theano was pivotal in accelerating the adoption and research of deep learning by combining mathematical rigor, computational efficiency, and GPU acceleration, marking a critical milestone in the evolution of AI tools

Caffe (2013)
Caffe, short for Convolutional Architecture for Fast Feature Embedding, is an open-source deep learning framework developed at the Berkeley Vision and Learning Center (BVLC) by Yangqing Jia in 2013 during his PhD at UC Berkeley. Designed with expressiveness, speed, and modularity as primary goals, Caffe quickly became a popular choice for researchers and practitioners working on convolutional neural networks (CNNs) and other deep learning models, especially in computer vision. The framework allows users to define, train, and deploy deep networks using configuration files without hard-coding, which encourages easy experimentation and innovation [46].

Caffe's architecture is built in C++ with a Python interface, providing flexibility and performance, including seamless switching between CPU and GPU computations by setting a single flag. It supports a wide range of neural network components such as convolutional, fully connected, and recurrent layers, and integrates NVIDIA's cuDNN library for GPU acceleration, enabling it to process over 60 million images per day on a single NVIDIA K40 GPU. Caffe was among the fastest convolutional network implementations available, making it highly suitable for both academic research experiments and industrial-scale deployments in vision, speech, and multimedia.

Mathematically, Caffe supports typical building blocks of deep learning: convolutions, pooling, activation functions, normalization, dropout, and backpropagation for gradient-based optimization. It leverages an efficient computation graph with layer-wise modularity, allowing various networks to be constructed by composing simple, reusable components. Its efficiency allowed state-of-the- art architectures like AlexNet and GoogleNet to be implemented and evaluated easily, accelerating breakthroughs in image classification and other domains [47].

The development of Caffe marked a significant milestone in deep learning research by emphasiz- ing performance, modularity, and ease of use, fostering an extensive community of contributors and users. Although its direct development waned after 2018 with the advent of newer frameworks like TensorFlow and PyTorch, Caffe's core ideas and implementations live on, influencing modern deep learning tools. It remains important historically for accelerating the adoption and advancement of CNNs in computer vision and AI research [48]

Tensor Flow (2015)
TensorFlow is an open-source deep learning framework developed and released by the Google Brain team in 2015 [49]. It emerged as the successor to Google's earlier proprietary system, DistBelief, and was designed to provide a flexible, scalable, and efficient platform for implementing machine learning and deep learning models. TensorFlow adopts a dataflow programming model where computations are expressed as stateful dataflow graphs consisting of nodes representing operations and edges representing multidimensional data arrays called tensors, from which the framework derives its name. Its architecture supports distributed computing and seamless deployment across CPUs, GPUs, and Google's custom Tensor Processing Units (TPUs), enabling training and execution of complex neural networks on diverse hardware from desktops to large server clusters and mobile devices.

Initially released under the Apache 2.0 open-source license, TensorFlow aimed to accelerate AI research and democratize access to powerful machine learning tools. Since its launch, TensorFlow has become the most popular and widely adopted deep learning framework globally, thanks to its combination of computational efficiency, comprehensive ecosystem, and extensive community support. It includes modules for a range of AI tasks such as model training, evaluation, prediction, visualization, and deployment. TensorFlow's Keras API integration, introduced from version 2.0 onwards, further simplified neural network programming by offering a high-level, user-friendly interface.

Mathematically, TensorFlow enables the construction of flexible computational graphs that repre- sent complex tensor operations. These graphs support automatic differentiation, a critical component for efficient backpropagation used in training deep neural networks. The system's parallel and dis- tributed design leverages optimization techniques and low-level integration with hardware drivers to maximize throughput and reduce training time. Its layered architecture abstracts details of hard- ware acceleration and memory management, allowing researchers and developers to focus on model innovation rather than implementation complexity.

TensorFlow's impact extends beyond academia and research labs into industry, powering applica- tions from image recognition to natural language processing and autonomous systems. Its continuous development by Google and an active open-source community ensures it remains a cornerstone of AI advancement and deep learning innovation worldwide [50].

Keras (2015)
Keras, introduced in 2015 by François Chollet, a Google engineer, is an open-source, high-level neural network API designed to simplify the development and experimentation with deep learning models. Developed as part of the ONEIROS research project (Open-ended Neuro-Electronic Intelligent Robot Operating System), Keras was motivated by the desire to make deep learning more accessible, modular, and user-friendly, targeting rapid prototyping and easy experimentation in neural network design [51]. Unlike low-level frameworks that require detailed knowledge of tensor operations and computational graphs, Keras offers a clean, intuitive API centered around building models layer-by- layer, which greatly reduces the cognitive load on developers.

Initially, Keras was independent, capable of running on multiple backend engines including TensorFlow, Theano, and Microsoft Cognitive Toolkit (CNTK). Starting from version 2.4, it was integrated into TensorFlow as its official high-level API through the tf.keras module, leveraging TensorFlow's computational efficiency while maintaining its signature simplicity. The modular design of Keras allows users to build complex neural network architectures from simple building blocks like layers, activation functions, optimizers, and loss functions, all customizable and extensible. Its design philosophy prioritizes ease of use without sacrificing flexibility, making it popular among researchers, students, and industry practitioners [52].

Mathematically, Keras encapsulates standard deep learning components such as dense layers (fully connected), convolutional layers, recurrent neural networks (LSTM, GRU), and allows automatic differentiation and backpropagation via its backend. It supports a range of optimization algorithms like stochastic gradient descent and Adam, and facilitates integration of custom loss functions or metrics. Keras's abstraction thus enables seamless experimentation with complex architectures while handling the underlying tensor operations and graph optimizations.

Keras's influence is profound as it lowered barriers to entry into deep learning research and development, accelerated experimentation cycles, and facilitated widespread adoption of neural networks. Its integration into TensorFlow and continued development positions Keras as a cornerstone tool in modern AI pipelines, beloved for balancing simplicity and power. François Chollet's original vision to democratize deep learning technology continues to resonate as Keras scales to support new frameworks and hardware.

Py Torch (2016)
PyTorch is a dynamic, open-source deep learning framework developed by Facebook AI Research (FAIR) and first released in 2016. Created as a successor to the Lua-based Torch framework, PyTorch was designed to provide researchers and developers with a more intuitive, Pythonic environment for building neural networks. By adopting a dynamic computational graph architecture—often described as define-by-run—PyTorch enabled program execution to be flexible and adaptable, allowing models to be modified on the fly during training or inference, which significantly simplified experimentation and debugging compared to static graph frameworks like TensorFlow [53].

The development team at FAIR, led by Soumith Chintala among others, focused on usability and seamless integration with the Python data science ecosystem. PyTorch supports accelerated tensor computation on CPUs and GPUs, automatic differentiation via the autograd system, and a rich set of predefined neural network components. In 2018, PyTorch 1.0 was released, merging the research- oriented PyTorch with production-ready Caffe2, unifying flexibility and scalability to streamline deployment from prototype to industrial applications.

Mathematically, PyTorch enables efficient representation and optimization of neural networks through dynamic construction and traversal of computation graphs. It supports standard deep learning operations, including convolutions, recurrent layers, and various activation and loss functions, combined with optimization algorithms like stochastic gradient descent and Adam. The autograd engine automates gradient computation via reverse-mode differentiation, integral to backpropagation training.

PyTorch's flexible interface, broad community support, and comprehensive tools have made it the framework of choice in AI research laboratories, academic settings, and increasingly in production environments. Its ecosystem now encompasses libraries for vision (TorchVision), natural language processing (TorchText), reinforcement learning, and more, while its adoption by major tech companies highlights its critical role in advancing AI innovation. The establishment of the PyTorch Foundation under the Linux Foundation in 2022 formalized its open governance, ensuring ongoing development guided by an industry consortium representing leading technology enterprises [54].

PyTorch's prominence stems from blending the agility needed for research with the demands of production, making it a flagship deep learning platform that continues to evolve at the forefront of AI development.

Cntk (2016)
Microsoft Cognitive Toolkit (CNTK), originally unveiled in 2016, is an open-source deep learning framework developed by Microsoft primarily for scalable machine learning across multiple GPUs and distributed systems. The roots of CNTK trace to internal Microsoft research needs, especially for accelerating speech and language processing projects—including models behind the Cortana virtual assistant and Bing web ranking. CNTK was created with a focus on performance, efficiency, and flexibility, aiming to democratize robust AI tools by making them available to researchers and practitioners through open-source licensing. Initially released with a proprietary scripting language known as BrainScript, CNTK quickly evolved to offer high-level APIs for Python, C++, and later C#, broadening its accessibility [55].

The framework models neural networks as computational graphs, allowing for intuitive rep- resentation and manipulation of feed-forward, convolutional, and recurrent architectures. CNTK supports automatic differentiation (backpropagation) and includes optimized readers for efficient handling of sparse and dense data in machine learning tasks. Its design enables parallelization and distributed learning—even over thousands of GPUs—utilizing innovations such as 1-bit stochastic gradient descent for highly efficient cross-node communication. Mathematically, CNTK enables state-of-the-art model training with rigorous implementation of deep learning constructs, from matrix and tensor operations to symbolic recurrent loops and multi-server parallelization. In addition to standard optimization algorithms, it supports advanced gradient propagation and memory sharing to maximize hardware utilization. A hallmark of CNTK is its seamless scalability, facilitating practical training of large neural models for image, speech, text, and time-series data.

CNTK found widespread use within Microsoft and externally, distinguishing itself by pioneering large-scale, efficient model training and offering strong integration with Windows and Linux environ- ments. Although its prominence has declined in favor of frameworks like TensorFlow and PyTorch, CNTK's technical innovations remain influential in large-scale machine learning infrastructure. Its last major release, version 2.7, supports ONNX interoperability and legacy AI projects, with many concepts and optimization strategies contributing to modern AI software engineering [56].

Mx Net (2015)
MXNet, officially known as Apache MXNet, is a scalable, open-source deep learning framework that was first released in 2015 and widely adopted by Amazon Web Services (AWS) as its preferred deep learning library. MXNet originated as a joint research effort between the University of Washington and Carnegie Mellon University, with key contributions from Carlos Guestrin and collaborators. The framework quickly gained traction in the industry for its unique combination of flexibility, efficiency, and distributed computing capabilities, making it ideal for both academic research and enterprise-scale production environments [57].

MXNet's architecture is distinguished by its hybrid programming model, seamlessly integrating both symbolic (static, declarative) and imperative (dynamic, Pythonic) approaches. Users can describe complex neural networks as computation graphs—benefiting from graph-level optimizations for memory and performance—or operate directly with tensor computations for maximum flexibility and debugging ease. A core innovation is the dynamic dependency scheduler that automatically parallelizes computation over CPUs and GPUs, and efficiently scales across multiple devices and nodes using a distributed parameter server for fast synchronization and data exchange.

Mathematically, MXNet supports building deep neural networks for supervised and unsuper- vised learning, including convolutional neural networks (CNNs), recurrent models (LSTMs, GRU), and advanced architectures. Its symbolic execution abstractions facilitate automatic differentiation and graph optimization for backpropagation, while imperative APIs offer intuitive manipulation of NDArray tensors. MXNet's scalability allows near-linear compute performance increases as GPU and CPU resources are added, which has been a decisive factor in handling large datasets and sophisticated models in production-scale applications.

MXNet's extensive language support—including Python, R, Scala, Julia, and more—and com- patibility with major cloud platforms like AWS, Microsoft Azure, and edge devices, further fueled its adoption. In 2017, MXNet became an Apache Top-Level Project, recognized for its robust governance and active community contributions. Despite its declining development activity due to industry shifts toward frameworks like PyTorch and TensorFlow, MXNet's legacy persists through its innovations in multi-language support, scalability, and cloud-native design [58].

Today, MXNet serves as an important case study in deep learning framework design, underpin- ning many of Amazon's AI services and continuing to influence large-scale neural network research and deployment strategies.

Deep Mind Alpha Go (2016)
DeepMind AlphaGo, developed by DeepMind (a subsidiary of Alphabet/Google), made history in 2016 as the first artificial intelligence system to defeat a reigning world champion in the ancient board game of Go—a feat long considered out of reach for machines due to Go's astronomical complexity and intuitive nature. Go presents a vast search space—estimated at 1017010170 possible board states—rendering brute-force search or traditional rule-based AI approaches inadequate. AlphaGo's innovation lay in integrating deep neural networks with advanced reinforcement learning and search techniques, bridging the gap between human-like intuition and rigorous calculation [59].

The AI's core architecture combined two deep neural networks: the policy network, trained using both supervised learning on millions of human expert moves and subsequently refined with reinforcement learning through self-play, proposes promising moves; and the value network, trained to predict the winner from a given board position, guides the evaluation of Go states. These networks underpin a massively parallelized Monte Carlo Tree Search (MCTS) algorithm, which simulates future play sequences and strategically explores the most relevant lines of play, prioritizing actions the policy network deems most probable and evaluating states through the value network. The networks themselves are convolutional neural systems with millions of parameters, capable of automatic feature extraction from raw board representations.

AlphaGo's training began by imitating human play from a large corpus of expert games, followed by intensive reinforcement learning wherein the system played countless games against versions of itself, refining its strategy far beyond human knowledge. The reinforcement learning step applies the policy-gradient approach, adjusting network weights to maximize board position values, guided by rewards based on game outcomes. Through this, AlphaGo developed powerful "intuition" for high-level play, even surprising expert players by inventing novel, creative moves.

AlphaGo's triumph over Lee Sedol in 2016 inspired the AI world, marking a paradigm shift in what reinforcement learning and neural networks could achieve, especially when combined with powerful computational resources and innovative training protocols. Its legacy extends beyond Go: the architecture and algorithms pioneered by AlphaGo have influenced domains as diverse as robotics, protein folding (AlphaFold), and resource optimization, demonstrating the transformative potential of deep reinforcement learning in solving problems of enormous complexity and strategic depth [60].

Open AI Gym (2016)
OpenAI Gym, introduced by OpenAI in April 2016, is a pivotal open-source toolkit specifically designed to advance the field of reinforcement learning (RL) by providing a standardized, extensible collection of simulation environments and benchmarks. The toolkit was developed to address two fundamental challenges in RL research: the lack of standardized environments for algorithm compari- son and the need for reproducible benchmarks to objectively evaluate and refine RL algorithms. By unifying how environments are structured and interacted with, OpenAI Gym accelerated the pace and rigor of RL research, enabling rapid prototyping, fair benchmarking, and cross-comparison of algorithms in both academia and industry [61].

Technically, Gym offers a Python API and a modular design, making it easy to create, wrap, or extend environments ranging from classic control systems (e.g., CartPole, MountainCar) to Atari 2600 games, physics-based robotics simulations (using MuJoCo, Box2D, or PyBullet), and custom, user-built tasks. Each environment adheres to a simple agent-environment interface: an agent observes a state, takes an action, receives a reward and a new state, and determines when an episode ends. This abstraction standardizes reinforcement learning experimentation, while the expanding collection of built-in environments accommodates both discrete and continuous action spaces, stochastic dynamics, and varying levels of task complexity. Gym's architecture supports seamless integration with popular deep learning frameworks like TensorFlow and PyTorch, allowing RL agents to leverage powerful neural networks for function approximation, policy learning, and value estimation.

OpenAI Gym also played a critical role in fostering an ecosystem around open RL research, inspiring the development of extensions like Gymnasium (its current community-maintained incarnation) and libraries for multi-agent RL, robotics, and benchmarking. Leading RL algorithms—from traditional Q-learning and SARSA to modern advancements in deep reinforcement learning—have been evaluated and reproduced using Gym's environments, which contributed to the field's rapid progress and reproducibility. Its influence extends to educational initiatives, practical applications in robotics, gaming, and AI modeling for complex sequential-decision problems [62].

In sum, OpenAI Gym's innovation lies in its standardization, extensibility, and community-driven design, which together underpin the contemporary landscape of reinforcement learning research, education, and application development

H2O.ai (2015)
H2O.ai, founded in 2012, is an open-source artificial intelligence and machine learning company with a vision of democratizing AI for individuals and businesses. Its most recognized platform, H2O, and subsequent AutoML offerings emerged as early leaders in delivering powerful, accessible machine learning solutions suitable for practitioners with varying technical expertise. The core H2O platform, commonly called H2O-3, was developed to provide scalable, distributed machine learning—capable of handling massive datasets in memory and natively integrating with Big Data frameworks like Hadoop and Apache Spark. This foundation made H2O widely used in both academic research and industry, thanks to its extensibility and high computational performance [63].

The introduction of H2O AutoML first released in 2017—propelled the H2O platform into the era of automation. AutoML lowers the barrier to entry for building machine learning models by automat- ing processes such as data preprocessing, feature engineering, model selection, hyperparameter tuning, and generation of stacked ensemble models. Users can build and deploy high-quality predictive models with minimal coding using intuitive APIs in R, Python, Java, and Scala, or the point-and-click H2O Flow web GUI. Under the hood, H2O AutoML efficiently conducts randomized model searches and ensemble learning to create a ranked leaderboard of models, balancing predictive accuracy with computational efficiency. Its distributed "H2O Cluster" architecture allows scaling across multi-node, multi-core environments, suitable for both on-premise and cloud deployments [64].

H2O.ai distinguishes itself through robust community participation and open-source ethos, along with features emphasizing interpretability, transparency, and regulatory compliance—a cornerstone for finance, healthcare, and high-stakes domains. Its Driverless AI product takes automation further, leveraging advanced techniques to build, optimize, and explain models automatically, making state-of- the-art AI more accessible and reliable.

With its blend of automation, scalability, and interpretability, H2O.ai has solidified its influence in the AI and data science landscape, empowering organizations of all types to leverage AutoML for fast, efficient, and explainable model building and deployment.

Generative & Pretrained Model Era (2017–2020)
Transformer Architecture (2017)
The Transformer architecture, introduced by Vaswani et al. in the landmark 2017 Google paper "Attention Is All You Need," revolutionized the field of natural language processing, sequence mod- eling, and—ultimately—became the foundation for modern large language models (LLMs). Prior to Transformers, neural sequence modeling was dominated by recurrent neural networks (RNNs) and long short-term memory (LSTM) architectures, which processed input data in order, making parallelization difficult and struggling with long-range dependencies [65].

The breakthrough of Transformers was their reliance solely on attention mechanisms—specifically, self-attention—to process inputs in parallel and capture context dependencies across entire sequences, regardless of their position. The model consists of an encoder and a decoder built from stacks of identical layers, each featuring multi-head self-attention and feedforward neural networks. Self- attention computes attention scores for each element, allowing the model to dynamically focus on the most relevant parts of input sequences when deriving contextual representations. Multi-head attention enables the model to consider multiple representation subspaces simultaneously, dramatically enhancing its capacity to encode nuanced relationships within data [66].

Mathematically, for a sequence of input vectors X, self-attention computes output vectors by weighting the importance of every other element using attention scores derived from scaled dot products. This mechanism is expressed as:

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

where Q, K, and V are the query, key, and value matrices derived from X, and dk denotes the dimensionality of the keys. This structure enables both efficient parallel computation and clear mathematical optimization, in contrast to the iterative nature of RNNs.

The impact of Transformer architecture has been enormous. It forms the backbone of all subse- quent foundational models in NLP and beyond, such as Google's BERT, OpenAI's GPT series, and many others [67]. These models deliver unprecedented performance in understanding and generating human language, powering everything from search engines to chatbots, content summarizers, and coding assistants. The ability to scale Transformers—by increasing their depth, width, and training datasets—led directly to the explosion of LLMs, with models reaching billions of parameters and previously unattainable capabilities. Research since 2017 has extended these principles to computer vision, speech, and multimodal AI, confirming Transformers as the defining innovation of the current era of machine learning and artificial intelligence.

Bert (2018)

BERT (Bidirectional Encoder Representations from Transformers), released by Google in 2018, marked a major leap in contextual language understanding and has become the cornerstone for many modern natural language processing (NLP) applications. Developed by Jacob Devlin and colleagues, BERT introduced the idea of deep, bidirectional pre-training that considers both the left and right context of every word in a sentence at every layer—unlike previous models (such as word2vec, GloVe, or even unidirectional Transformer-based models like GPT) that process language in a single direction or with only shallow bidirectionality [68].

Built on the encoder segment of the Transformer architecture, BERT leverages self-attention to create contextual embeddings, enabling it to capture subtle meanings of words based on the entire sentence. The original BERT model was released in two standard sizes: BERT_BASE (12 layers, 12 attention heads, 110M parameters) and BERT_LARGE (24 layers, 16 attention heads, 340M parameters). Its bidirectional context modeling allows BERT to resolve ambiguous words with high accuracy, making it significantly more powerful for tasks like question answering, sentiment analysis, and coreference resolution.

BERT's training involves two novel self-supervised objectives:

- Masked Language Modeling (MLM): Randomly masks words in a sentence and trains the model to predict them using the surrounding context, fostering deep bidirectional understanding [69].
- Next Sentence Prediction (NSP): Trains the model to predict whether a given sentence logically follows another, enhancing its ability to grasp sentence relationships and discourse structure.

After pre-training on massive corpora, BERT can be fine-tuned with just an additional output layer for virtually any NLP task, and achieves state-of-the-art results across a broad spectrum of benchmarks. Its open-source release sparked a surge of innovation, spawning a family of models (e.g., RoBERTa, ALBERT, DistilBERT) and fundamentally changing the standard paradigm for NLP: from training task-specific models from scratch to leveraging large, pre-trained "foundation" language models and adapting them for downstream tasks.

BERT's adoption by Google Search in 2019 brought its impact to billions of users worldwide, dramatically improving search query understanding for over 70 languages. BERT's introduction established the pre-train-then-fine-tune model as the dominant approach to language understanding and catalyzed the development of ever-larger and more powerful language models, ushering in the post-BERT era in AI [70].

GPT-2 (2019)

GPT-2 (Generative Pre-trained Transformer 2), released by OpenAI in February 2019, was a landmark breakthrough in generative language modeling. Building directly on its predecessor GPT-1, GPT-2 featured a dramatic leap in scale—with up to 1.5 billion parameters—and showcased the power of unsupervised learning on massive datasets. Trained on the WebText corpus (8 million web pages, 4˜0GB), GPT-2 demonstrated the ability to produce multi-paragraph, coherent, and contextually relevant text with only a simple prompt, making it suitable for tasks such as story generation, translation, question answering, summarization, and even composing poetry [71].

Its architecture is a decoder-only Transformer—a stack of multihead self-attention and feedfor- ward layers—operating autoregressively (predicting the next word given preceding text). GPT-2's design allows it to model long-range dependencies and generate fluid, adaptive prose. The model's "chameleon-like" flexibility enabled strong zero-shot and few-shot learning: GPT-2 could perform a variety of tasks without any task-specific training data, simply by interpreting instructions given in natural language [72].

OpenAI opted for a staged release, initially limiting public access to smaller models before fully releasing the largest 1.5-billion-parameter version in November 2019. This cautious approach was due to concerns over misuse, such as generating convincing fake news, spam, or impersonation content—GPT-2 was one of the first AI models to raise broad public debate over the ethical risks of advanced text generation technology. Despite these concerns, GPT-2 rapidly became a research baseline, demonstrating that scaling up Transformers led to dramatic improvements in text generation quality and general-purpose language understanding [73].

GPT-2's impact is immense: it inspired a wave of larger, more capable models (GPT-3, GPT-4), catalyzed the development of conversational AI, creative writing tools, and powerful text-based assistants, and underscored the paradigm of language model pre-training followed by task-specific adaptation. Its code and checkpoints remain widely used in research and industry, and its success marked the beginning of the large language model (LLM) revolution [74].

Style GAN (2019)
StyleGAN, introduced by Nvidia researchers and made source-available in February 2019, is a groundbreaking generative adversarial network (GAN) architecture that set a new benchmark in ultra- realistic image synthesis—most notably for generating convincing portraits of non-existent human faces. Building upon previous GAN advancements, especially Nvidia's Progressive GAN (ProGAN, 2017), StyleGAN's innovation lies in its ability to provide fine-grained control over visual attributes ("styles") at different levels of image abstraction, from the coarse (face shape, pose) to the fine (hair texture, freckles, wrinkles).

The core of StyleGAN architecture uses a mapping network that transforms a random latent vector into a set of "style" vectors [75]. These style vectors are injected at multiple layers of the generator, allowing for highly controllable synthesis of image features. This "adaptive instance normalization" based mechanism means users can smoothly interpolate features—mixing, blending, and manipulating aspects of generated faces with remarkable realism. Standard GAN training is retained, where a generator learns to create images and a discriminator learns to distinguish real from fake, iteratively improving the realism of outputs. [76]

StyleGAN's technological leap became widely recognized through the viral website "This Person Does Not Exist," which showcased the ability to generate endless, lifelike human faces with each refresh. Its impact reverberated through both creative and scientific domains: StyleGAN is used in art, game graphics, synthetic data generation, and education about media authenticity. Following the original release, Nvidia improved the method with StyleGAN2 (2020), which removed visual artifacts and enhanced image quality, and StyleGAN3 (2021), which solved "texture sticking" and delivered more alias-free generation—further refining the consistency of generated details.

Mathematically, StyleGAN's generator employs convolutional neural layers where style vectors modulate normalization parameters. This enables controlled variation in features and "style mix- ing"—a capability earlier GANs lacked. The resultant images regularly surpass previous models in realism as measured by metrics such as the Fréchet Inception Distance (FID).

In summary, StyleGAN's introduction opened new possibilities in visual synthesis, enabling ultra-realistic image generation, fine feature manipulation, and creative exploration. Its releases and iterative improvements remain critical milestones in the history of AI-generated imagery and machine creativity [77].

T5 (2019)
T5 (Text-to-Text Transfer Transformer), introduced by Google Research in 2019, is a transformative language model that unified how natural language processing (NLP) tasks are framed and solved. Unlike earlier models that treated tasks such as translation, summarization, and question answering with custom architectures or approaches, T5 proposed a "text-to-text" framework: every problem is cast as converting an input text string into an output text string, regardless of the underlying NLP task [78].

Built on the full encoder-decoder Transformer architecture, T5 processes input text using the encoder and generates output text with the decoder, leveraging the power of attention mechanisms throughout. The unifying principle is to prepend tasks with special instruction-like text prefixes (e.g., "translate English to German: That is good") and train the model [79]end-to-end on a diverse range of tasks using the same architecture and loss function. This provides a consistent interface and allows multitask learning, where the same model can be fine-tuned or prompted for a wide array of downstream applications—including translation, summarization, classification, question answering, and more [80].

For pre-training, T5 uses a large, high-quality dataset called the Colossal Clean Crawled Corpus (C4), containing hundreds of gigabytes of web-scraped English text. The self-supervised learning objective is "span corruption," where random spans of text are replaced with sentinel tokens and the model learns to reconstruct the missing content, enabling deep contextual and compositional language understanding. T5's vocabulary is built using SentencePiece tokenization, allowing coverage of multiple languages and efficient handling of rare or out-of-vocabulary words [81].

By recasting every NLP task as a text transformation, T5 not only simplified the training and deployment process but also delivered state-of-the-art performance across a diverse array of language benchmarks. Its influence is seen in subsequent developments in "instruction tuning" and general- purpose, instruction-following LLMs, which still trace their roots to T5's unified approach [82].

T5 represents a major evolution in NLP thinking, demonstrating the power of casting all language problems into a single, flexible text-to-text mold while leveraging large-scale transfer learning [83].

Fast AI (2018)
FastAI, launched in 2018 by Jeremy Howard and Sylvain Gugger, is an open-source deep learning library built atop PyTorch that is designed to make state-of-the-art machine learning accessible to beginners and experts alike. FastAI abstracts and automates many of the complexities of deep learning, enabling rapid experimentation, intuitive workflows, and best practices by default, while still allowing for full access to underlying PyTorch capabilities [84].

The library introduces a layered API:

- High-level: Functions for common deep learning tasks (vision, text, tabular, time series, collabo- rative filtering) that minimize code and domain-specific knowledge requirements.
- Mid-level/Low-level: Modular building blocks that let advanced users customize model architec- tures, training strategies, and data preprocessing pipelines.
- Core FastAI concepts include:
- Data Block/Data Loaders: Cleanly structured tools for scal-

able, flexible data preprocessing and loading.
- Learner: Encapsulates a complete model training pipeline—bringing together data, network architecture, training/evaluation logic, and reporting.
- Built-in Best Practices: Automated data augmentation, mixed precision training, transfer learning integration, and state-of-the-art optimizers simplify robust model development.

Mathematically, FastAI leverages PyTorch for tensor operations, differentiation, and GPU accel- eration. Its API extends PyTorch's flexibility with powerful abstractions that enable researchers to prototype, train, and deploy models with less boilerplate and more focus on innovation.

FastAI's vibrant ecosystem is complemented by comprehensive courses, documentation, and a large community, which has contributed to making cutting-edge AI techniques approachable for practitioners, educators, and researchers across the world. It has been instrumental in democratizing deep learning education and practice, accelerating the adoption of PyTorch and modern deep learning best practices in both industry and academia [85].

Allen NLP (2018)
Allen NLP, launched in 2018 by the Allen Institute for Artificial Intelligence (AI2), is an open- source research library built on top of Py Torch that is dedicated to advancing natural language processing (NLP) research and applications. Allen NLP addresses the common challenges faced by NLP researchers—such as reproducibility, extensibility, and ease of experimentation—by providing reusable building blocks, modular data pipelines, and configuration-driven experiment management [86].

Key architectural features of Allen NLP include:
- Py Torch Foundation: Leveraging Py Torch's dynamic computation graphs, enabling flexible model design and intuitive debugging.
- Modular Components: Reusable modules for tokenization, data reading, embedding, encoding, and pre/post-processing, which allow rapid prototyping and efficient pipeline construction.
- Declarative Configurations: JSON or Python-based experiment configurations, making it easy to define, reproduce, and share experimental workflows, models, and hyperparameters.
- Reference Implementations: High-quality models for a variety of NLP tasks, such as semantic role labeling, textual entailment, question answering, and named entity recognition, help users benchmark and extend cutting-edge research methods.
- Flexible Data API: A "Field" and "Instance" abstraction allows unified and efficient handling of
- diverse NLP data structures, such as sequences, spans, and trees, with automatic sorting, batching, and padding.

Allen NLP is especially valued in the NLP research community for streamlining new model development and fostering reproducibility. It also comes with tools for visualization, evaluation, and integration of pre-trained models. Its APIs and pipelines have made it a popular choice for both academic research and real-world NLP productization [87].

Applications range from text classification, semantic parsing, and coreference resolution to infor- mation extraction and knowledge graph construction. Allen NLP continues to be actively developed and widely adopted, facilitating rapid advances in NLP research and supporting open science through a thriving community and extensive documentation.

Hugging Face Transformers (2019)
Hugging Face Transformers, launched in 2019, rapidly became the most influential open-source platform and library for accessing, sharing, and deploying pre-trained models in modern natural language processing (NLP). Initially, Hugging Face was focused on chatbot development, but its founders soon recognized the profound potential—and the community's need—for a unified hub that would make cutting-edge transformer models (beginning with BERT, GPT-2, and others) easily accessible to all practitioners and researchers[88] . The result was the Transformers library—a consistent API and repository that supports Py Torch and TensorFlow, and enables seamless downloading, fine- tuning, and deployment of models for tasks such as text classification, question answering, translation, summarization, and more[89].

This platform's breakthrough was to "democratize" access to the most powerful models [90] resulting from the "transformer revolution," which had previously been confined to specialized labs or required significant engineering expertise to reproduce. In the Hugging Face ecosystem, anyone can load a model with a line of code, experiment interactively, and share improvements or new models via the hosted Model Hub. It soon evolved from supporting a handful of models to hosting thousands—including nearly all major transformer-based architectures like RoBERTa, T5, DistilBERT, XLNet, and later GPT-3, BLOOM [91] , and large multimodal models. Alongside the core library, Hugging Face released companion tools for tokenization, dataset management, and evaluation, making it not only a toolkit for inference but a full-stack environment for research, production, and benchmarking. The impact on NLP can hardly be overstated. By radically lowering barriers to entry, Hug- ging Face enabled rapid experimentation, broad collaboration, and the sharing of reproducible re- sults—accelerating progress on major benchmarks, downstream applications, and even in languages and domains with less well-funded research. Its open-source ethos cultivated a global community of contributors and users ranging from machine learning engineers and academic labs to major tech companies and startups. As a result, Transformer-based methods became standard practice for a huge swath of industry and academia [92].

Moreover, Hugging Face Transformers now plays a critical role in shaping the landscape of generative AI, powering conversational agents, language generators, document classifiers, and bio-in- formatics solutions. It bridges research and production, providing high-level interfaces suitable for non-experts and deeply customizable options for advanced users. Its effective blend of technical excel- lence, user experience, and community-driven development has solidified its status as the "GitHub for machine learning models"—a foundational resource in today's AI ecosystem, and a major force in the continuing evolution of large language models and intelligent systems [93].

Multimodal & Generative AI Explosion (2020–2023)

GPT-3 (2020)

GPT-3 (Generative Pre-trained Transformer 3), released by OpenAI in 2020 [94]. marked a major milestone in the evolution of generative AI. With a staggering 175 billion parameters, GPT-3 dwarfed all previous language models—being over a hundred times larger than GPT-2—and signaled a new era of artificial intelligence powered by scale and transfer learning. Like its predecessors, GPT-3 is a decoder-only Transformer model, built on deep neural network layers that leverage self-attention to analyze and synthesize input text. It was trained on a vast corpus—over 45 terabytes of diverse text from Common Crawl, Wikipedia, books, and scientific articles—allowing it to attain fluid, humanlike language abilities.

The most remarkable advance with GPT-3 is its zero-shot and few-shot learning capabilities. With just minimal or no task-specific fine-tuning, GPT-3 can generate text, answer questions, trans- late languages, summarize, solve arithmetic problems, write computer code, and adapt to complex prompts—all by interpreting context and examples given at inference time. This “prompt engineering” approach allows users to unlock new behaviors from a single model without retraining, a radical leap compared to previous generation systems. The model’s context window of 2048 tokens enable understanding and retention of lengthy passages, contributing to coherent multi-paragraph output [95].

GPT-3’s emergence resulted in widespread impact across industries. It powers chatbots, creative writing assistants, educational tools, content automation systems, and is the core technology behind OpenAI’s later product ChatGPT (initially based on GPT-3.5, later upgraded to GPT-4). Its performance on standardized NLP benchmarks set new records, and its general-purpose text generation was so convincing that many human evaluators struggled to distinguish its writing from authentic human prose [96]. The scale of GPT-3 also raised concerns about ethical risks, misinformation, and the social responsibility of deploying powerful language models, prompting debates that continue as models grow even larger and more capable.

Licensing and API access to GPT-3 reflected both excitement and caution; Microsoft secured exclusive rights to the underlying model, while others access it via OpenAI’s cloud API. In the wake of GPT-3, the field experienced a surge of competitive innovation—spurring subsequent large models from Google, Meta, and open initiatives like EleutherAI [97].

CLIP (2021)

CLIP (Contrastive Language–Image Pretraining), introduced by OpenAI in 2021, represents a key milestone in multimodal AI by fundamentally bridging the world of images and natural language. Unlike classical computer vision models, which require extensive supervised training for each specific task, CLIP learns to associate images and text in a unified representation space through a process called contrastive learning. The model was trained on an unprecedented scale—400 million image–text pairs sourced from the internet—allowing it to develop a broad and general understanding of visual concepts as expressed in natural language [98].

The technical architecture of CLIP consists of two separate neural networks: a vision encoder (typically a Vision Transformer or ResNet) and a text encoder (usually based on a Transformer). Both encoders map their respective inputs—an image and a piece of text—to high-dimensional embeddings in a shared feature space. During training, CLIP maximizes the cosine similarity between embeddings for matching image–caption pairs and minimizes it for mismatched pairs. The outcome is a model that can, given an image and a set of candidate text descriptions (or vice versa), identify which text best matches the image—even for objects or situations it has never seen before [99].

What makes CLIP extraordinary is its "zero-shot" capability: without any further task-specific training, it can classify images, retrieve relevant images given text queries, generate image captions, and more, simply by leveraging the rich relationships embedded in its multimodal representation. For instance, CLIP can instantly label previously unseen images by choosing from a list of natural language prompts, far surpassing the flexibility of models reliant on tightly controlled class labels or datasets. Its approach to generalization also inspired the architecture of numerous AI systems powering text-to-image synthesis, content moderation, and robust semantic search [100].

CLIP’s release was also transformative at the ecosystem level. It became an essential building block for more advanced multimodal models, including guidance systems for generative models like DALL-E and Stable Diffusion. In industry and research, CLIP powers applications ranging from search engines and recommendation systems to digital art, content filtering, and the investigation of neural network interpretability (such as "multimodal neurons"). Its public release has enabled wide experimentation and rapid progress in building truly general-purpose, flexible AI capable of connecting language and vision [101].

DALL·E (2021)

DALL·E, unveiled by OpenAI in January 2021, is a pioneering deep learning model designed to generate novel images from text prompts—a capability that captured the world’s attention and redefined the potential of generative AI. Building on the success of large language models like GPT-3, DALL·E employs a Transformer-based architecture and leverages concepts from autoregressive models and variational autoencoders (VAEs). It was trained on hundreds of millions of images–text pairings collected from the internet, allowing the model to synthesize strikingly creative and highly coherent visuals based solely on detailed natural language descriptions [102].

The technical breakthrough of DALL·E lies in representing both text and images as discrete tokens. During training, a discrete VAE compresses each image into a lower-dimensional grid of tokens, while the Transformer learns to model the joint distribution of text and image tokens as a single sequence [103]. When given a prompt, DALL·E generates the sequence of image tokens that, when decoded by the VAE, produces an image matching the semantic content of the text. This means a prompt like “an armchair in the shape of an avocado” will yield an entirely new image, blending previously unseen concepts and styles in photorealistic or artistic ways.

One of DALL·E’s most significant impacts is its seemingly limitless generative creativity and compositional reasoning. The model can merge disparate ideas, invent objects, adapt styles, or transform concepts much like an imaginative artist responding to verbal instructions. DALL·E's "zero- shot" ability—generating meaningful imagery for prompts it was never explicitly trained on—quickly found applications in art, design, education, and synthetic data generation [104].

Following the original DALL·E, OpenAI released DALL·E 2, which improved image quality and capability using advances in diffusion models guided by CLIP, OpenAI's multimodal text–image embedding network. These innovations established a new paradigm for generative models: text- to-image (and now text-to-video and text-to-3D) synthesis that democratizes visual creativity and speeds up workflows in media, design, and scientific visualization. Both the original DALL·E and its successors have become benchmarks, inspiring further development in open-source generative models and autonomous creative systems worldwide [105].

Stable Diffusion (2022)
Stable Diffusion, released in August 2022 by Stability AI in partnership with Comp Vis and Runway ML, transformed the generative AI landscape by making photorealistic image synthesis open-source and accessible to everyone—even those without enterprise-level hardware or budgets. Unlike previous proprietary models like DALL·E or Midjourney, Stable Diffusion's source code and model weights were openly licensed, allowing anyone to download, run, and modify the system for personal, academic, or creative use. This choice democratized AI-based image generation, fueling an explosion of community-driven innovation and applications across art, media, research, and industry [106].

Technically, Stable Diffusion is a latent text-to-image diffusion model. It works by first compress- ing images into a lower-dimensional latent space using an autoencoder, then employing a neural network to iteratively reverse random noise back into a coherent picture, guided by a textual prompt. The choice of working in latent space—as opposed to the full pixel space—allows for faster image generation and reduces computational requirements, enabling high-quality results even on consumer GPUs. For understanding prompts, Stable Diffusion incorporates OpenAI's CLIP text encoder, which maps natural language instructions to vector representations closely aligned with the visual domain, ensuring that generated images faithfully reflect the user's input [107].

Open-source accessibility led to massive adoption and rapid evolution. Creators and developers built custom interfaces, extensions, and plug-ins, while artists and designers gained powerful new tools for ideation, prototyping, and artistic expression. The flexibility to modify and fine-tune models spawned countless niche variants and tailored solutions, expanding the possibilities for commercial, educational, and creative uses. Stable Diffusion also prompted important conversations around ethics, copyright, and the future of creative work, particularly as human artists contended with new forms of digital art generation and content authenticity [108].

Culturally and technologically, Stable Diffusion is credited with "removing the doors from their hinges"—ushering in a new era where AI image generation became a grassroots phenomenon rather than a privilege for major tech companies. With billions of images created and a vibrant community fueling continued improvement, Stable Diffusion remains a backbone of generative art, synthetic media, and open research, pressuring proprietary competitors to open up and spurring advances throughout the AI field [109].

Midjourney (2022)
Midjourney, launched in early 2022, quickly established itself as one of the leading platforms for AI-driven art generation, harnessing the power of text-to-image synthesis to empower creators of all kinds. Founded by David Holz (also known for Leap Motion), Midjourney is an independent research lab whose eponymous software enables users to craft striking, imaginative visuals from simple text prompts. Unlike previous art generators focused primarily on photorealism or technical demonstration, Midjourney is distinguished by its painterly aesthetics, creatively blending real-world styles, artistic influences, and fantastic elements. Its model excels in fantasy scenes, stylized environments, and expressive character portraits a favorite among concept artists, designers, illustrators, and hobbyists [110].

Access to Midjourney is innovative: the platform operates primarily as a bot on Discord, where users type prompts and receive images instantly, sparking communal sharing, feedback, and collab- orative exploration. This workflow makes AI art creation interactive and social, driving the rapid growth of a global user base that has surpassed 16 million by late 2023. Midjourney offers continual updates—releasing improved algorithms every few months, each lifting artistic quality, coherence, prompt accuracy, and stylistic variety. By 2024, its web interface expanded accessibility, bringing AI visual generation beyond Discord's audience [111]. Under the hood, while Midjourney is speculated to use principles similar to latent diffusion models (as in Stable Diffusion), its proprietary technology and blend of artistic tuning set it apart. The system leverages high-quality image–text pairing, learning complex associations between language and visual style, and allowing highly customized creations based on user instructions. The result is a new form of visual ideation, letting professionals and enthusiasts prototype commercial art, create book illustrations, design product concepts, or simply explore creative possibilities, often in minutes [112].

Midjourney's impact stretches far beyond its technical achievements. Its accessible, community- centered approach to AI-generated art has helped disrupt traditional stock photography, lowered creative barriers, and brought sophisticated art generation to the masses. The platform has powered magazine covers, children's books, concept art, and even winning entries in digital art competitions. Users praise its ability to speed up brainstorming, explore unfamiliar styles, and develop visual narratives with unprecedented ease [113].

Whisper (2022)
Whisper, released by OpenAI in September 2022, is an open-source automatic speech recogni- tion (ASR) model that set a new standard for transcribing and translating audio across a diverse range of languages, accents, and audio environments. Its development was motivated by the need for robust, accurate, and scalable speech-to-text systems not just for English but for over

100 lan- guages worldwide—addressing limitations of earlier models that struggled with non-standard accents, domain-specific jargon, and noisy backgrounds [114].

At the heart of Whisper's approach is a weakly-supervised training pipeline and a sequence- to-sequence encoder-decoder Transformer architecture. The model takes as input a log-Mel spectro- gram—a time-frequency representation—of audio split into 30-second chunks, transforming it into a sequence of latent vectors that capture temporal, spectral [115], and semantic information. The decoder then generates transcriptions token by token, with the same model capable of language identification, transcription in the original language, translation into English, and phrase-level timestamps. This multifunctional design was achieved by training on an exceptionally large and diverse dataset of 680,000 hours of supervised audio-text pairs sourced from the web, which enabled the model to learn a wide variety of accents, recording conditions, and technical vocabularies.

Whisper's release as open source offered unprecedented benefits for developers, researchers, and practitioners. It provided not only high-accuracy English transcription but also robust multilingual transcription and speech-to-English translation without any extra fine-tuning. The model's versatility and ease of use sparked rapid adoption, enabling better accessibility solutions, improved voice interfaces, podcast and video transcription, language study tools, and facilitating research into more complex audio and speech applications [116]. Whisper's ability to generalize across languages, accents, and environments has made it a founda- tional technology in the global push for inclusive, AI-powered speech and language tools. Moreover, its transparent release set a benchmark for openness in AI development, supporting further innovation in speech recognition technology and its integration into both commercial and open-source projects [117].

ChatGPT (2022)
ChatGPT, released by OpenAI in November 2022, represents one of the most remarkable leaps forward for conversational artificial intelligence. Built initially on GPT-3.5 and later enhanced by GPT-4, ChatGPT marked the first time a large language model was fine-tuned and deployed specifically for natural dialogue with broad, everyday usability. Its launch was a watershed moment: millions adopted ChatGPT within days, integrating it into personal and professional routines for Q&A, drafting content, brainstorming, tutoring, technical assistance, and creative writing. The system's ability to provide coherent, contextually relevant, and human-like responses—often with nuanced understanding and personality—demonstrated the practical viability of large-scale conversational AI for general use [118]. ChatGPT's core advances derive from two pivotal techniques. First, the conversational interface was built to handle multi-turn interactions, enabling users to clarify, elaborate, and correct threads of dialogue in natural language. Second, the model was fine-tuned via reinforcement learning from human feedback (RLHF), aligning responses with human preferences, improving safety, and min- imizing bias and harmful outputs. This combination made ChatGPT surprisingly capable at not just answering questions, but engaging in back-and-forth exchanges, reasoning through ambiguity, admitting mistakes, and playfully challenging incorrect premises [119]. ChatGPT's underlying models evolved rapidly: starting with GPT-3.5, which offered strong general-domain language abilities, then moving to GPT-4 in March 2023, whose larger scale, im- proved alignment, and multimodal capabilities pushed comprehension and creativity even further. By 2024—2025, versions with even broader context windows, enhanced reasoning, and tool integration were released, enabling the system to use web search, interpret images, code, and analyze user data. Throughout, model architecture remained based on the Transformer paradigm, with progressive improvements in alignment, factuality, steerability, and reasoning depth [120].

The impact of ChatGPT is profound and ongoing. It has been widely adopted in customer service, education, content creation, programming, healthcare, entertainment, and more—changing workflows and user expectations about interacting with computers. Its ease of use—running in browsers, apps, and APIs—helped it reach billions and inspire a new wave of competition among tech companies [121].

Perhaps even more importantly, ChatGPT popularized prompt engineering and conversational design as skills, enabled researchers to probe both the powers and limitations of LLMs, and raised important ethical, legal, and social questions about synthetic dialogue, misinformation, and AI-assisted work [122].

Bloom (2022)
BLOOM (BigScience Large Open-science Open-access Multilingual Language Model), released in July 2022, is a landmark achievement in the movement toward open, collaborative, and transparent AI research. Developed over a year-long workshop by the BigScience collaboration—a vast consortium coordinated by Hugging Face and comprising more than a thousand researchers from around the world—BLOOM is one of the largest large language models ever made publicly accessible. With 176 billion parameters, BLOOM stands alongside proprietary models like GPT-3 and PaLM but distinguishes itself by distributing both its model weights and training data under permissive open licenses, allowing for unrestricted research, adaptation, and scrutiny worldwide [123].

The architecture of BLOOM is a transformer-based autoregressive language model, trained to generate text and code across 46 natural languages and 13 programming languages. Its multilingual proficiency was a crucial focus, making it the most widely inclusive open-source LLM of its time and a powerful tool for countering the English-centric bias seen in earlier AI models. To achieve this, BLOOM was trained on the diverse, transparent ROOTS corpus, containing nearly 1.6TB of text carefully sourced, documented, and cleaned to reflect not just the web but also underrepresented communities and linguistic groups [124].

Transparency extended far beyond the code and weights—the whole process, including engineer- ing decisions, training logs, and ethical deliberations, was carried out in public view on the Jean Zay supercomputer in France. This pioneering approach allowed researchers everywhere to follow, audit, and contribute at every step, setting a new gold standard for open science and reproducibility in the AI field [125].

BLOOM's contributions extend beyond its technical prowess.

By making such a model openly available, it democratized large-scale AI, enabling communities, universities, and smaller innova- tors—especially those from developing regions—to participate in frontier machine learning research without the heavy restrictions or costs associated with proprietary models. This has fostered not only novel applications and research but also greater global dialogue around bias, governance, multilingual access, and responsible AI development [126].

LLa MA (2023)
LLaMA (Large Language Model Meta AI), released by Meta (formerly Facebook) in early 2023, represents a major advancement in accessible large language model research. Developed as a suite of transformer-based models with parameter sizes ranging from 7 billion to 65 billion for the first generation (and up to 70 billion in subsequent versions), LLaMA was designed to rival performance benchmarks set by proprietary LLMs like GPT-3 and PaLM, but with a focus on efficiency, transparency, and open availability to the global research community [127].

The initial L La MA models were provided under a research license, and while they were intended for non-commercial use and distribution to approved academic applicants, their weights were rapidly disseminated online, catalyzing a burst of experimentation and fine-tuning by independent researchers worldwide. This "leak" incident, while controversial, contributed to the immediate formation of an active ecosystem: academics and developers built instruction-following variants like Alpaca (Stanford), Vicuna (LMSYS), and Koala, showing that state-of-the-art conversational AI could be achieved for relatively low cost and infrastructure by fine-tuning L La MA foundations [128].

Meta's follow-up, L La MA 2 (July 2023), further revolutionized access to advanced LLMs. L La MA 2 models—released in 7B, 13B, and 70B parameter sizes—offered not just foundational model weights but also chat-optimized, instruction-following variants with enhanced safety fine-tuning. Notably, L La MA 2's more permissive license enabled commercial use under certain conditions and included wide platform integration with partners like Microsoft and Hugging Face, dramatically expanding its adoption in industry and research. The release of Code L La MA, specialized for code generation and programming, further deepened its impact on technical and developer communities [129].

Technically, L La MA's success is often attributed to highly efficient model design, extensive multilingual and public dataset pretraining, and rigorous evaluation on diverse benchmarks. Its training datasets were constructed to optimize both language understanding and generation while making models compact enough for fine-tuning and inference on standard hardware, democratizing LLM research and customization worldwide.

L La MA's open availability has catalyzed a global movement of "open foundation" model devel- opment. Researchers and organizations now routinely create specialized, localized, and safety-aligned derivatives, pushing the boundaries of large language models while encouraging transparency, repro- ducibility, and broader participation in generative AI. At the same time, the L La MA family has raised important questions about responsible model sharing and open science, as the ease of adaptation brings both societal benefits and risks [130].

Gemini (Bard) (2023)
Gemini, unveiled by Google DeepMind in December 2023, is Google's most advanced multimodal AI model to date—purpose-built to natively understand, generate, and reason across text, images, audio, video, and code. Gemini was designed from the ground up as a multimodal system, meaning that, unlike previous models which often combined separately trained modules, Gemini's architecture was jointly trained on diverse data types. This unified approach enables the model to seamlessly process complex, interleaved sequences of language and visuals, handle elaborate reasoning tasks, and provide highly contextual, accurate responses to a broad array of queries involving multiple modalities [131].

The Gemini family offers several model sizes—Ultra, Pro, and Nano—optimized for different use cases, from high-performance research to on-device AI for smartphones. Notably, Gemini Ultra has set new records on over 30 major academic and industry benchmarks, including being the first model to surpass human expert performance on the Massive Multitask Language Understanding (MMLU) exam. Gemini excels at a wide range of tasks, such as reading and interpreting complex diagrams, infographics, and scientific documents; analyzing natural images and video; audio transcription and understanding; sophisticated mathematical and logical reasoning; and robust code generation spanning multiple programming languages [132]. Gemini's multimodal strength offers unique advantages in solving challenging questions that blend visual, linguistic, and logical information, making it highly effective for scientific discovery, education, content creation, and enterprise analytics. The system's architecture directly supports the ingestion of text, images, audio waveforms, and video frames, allowing for nuanced analysis and synthesis that bridges gaps between previously siloed data formats [133].

Gemini also powers Google's Bard chatbot (now rebranded as Gemini across many Google services), bringing enhanced generative AI to Search, Assistant, Android devices, and developer platforms like Google AI Studio and Vertex AI. Its natively multimodal reasoning, broad multilingual support, and ability to extract insights from massive datasets are driving innovation across consumer and enterprise applications [134].

Critically, Gemini isn't just a technical achievement—it marks the culmination of Google's efforts to bring together DeepMind and Google Brain, leveraging global collaboration and resources to set new standards for open, scalable, and responsible AI development. As Gemini continues to evolve—with version updates such as Gemini 1.5 and on-device deployment via Pixel phones—it is poised to define the next wave of general-purpose, highly capable, and accessible artificial intelligence [135].

Claude (2023)
Claude, released by Anthropic in March 2023, is a family of large language models engineered to push the state of conversational AI with a strong emphasis on safety, ethical alignment, and user control. Developed by a team of former OpenAI researchers, Claude's defining innovation is its use of "Constitutional AI"—a training approach that combines Reinforcement Learning from Human Feedback (RLHF) with a set of guiding principles or "constitution." This constitution, drawn in part from documents

like the UN Universal Declaration of Human Rights, directs the model to behave harmlessly and helpfully, relying not only on human feedback but on its own self-critiques and revisions. In practice, this method allows Claude to generate, review, and improve its own outputs while aligning with clear ethical standards, minimizing risks of bias, misinformation, and harmful responses [136].

Anthropic continually expanded Claude's capabilities, releasing successive versions that increased the maximum context window—up to 200,000 tokens, or roughly 500 pages of material. This enabled users to upload and process lengthy documents, and perform advanced summarization, document analysis, and text-based research tasks well beyond prior systems. With Claude 2 and beyond, the models could read, interpret, and assist with tasks using a diverse range of inputs, including PDFs and complex workflows.

Safety and reliability are central to Claude's design. The model demonstrates lower rates of hallucination and harmful output, and in recent updates, incorporates mechanisms capable of ending conversations in extreme cases of persistently abusive user interactions—reflecting Anthropic's com- mitment to both model and user "welfare." These capabilities, along with steerable tone, persona, and user feedback responsiveness, make Claude a uniquely user-friendly system for business, education, programming, creative writing, and data analysis [137].

Anthropic's approach has spurred significant debate on the balance between ethical safeguards and usability, as some critics claim the model's refusal to answer certain benign requests constitutes an "alignment tax." Nonetheless, the importance of transparency, privacy, and verifiable ethical alignment places Claude at the leading edge of responsible and trustworthy AI development [138].

The Current Multimodal & Agentic Era (2024–2025)

GPT-4 / GPT-4o (2024)

GPT-4 and its successor GPT-4o (the "o" stands for "omni"), released by OpenAI in 2024, are state-of-the-art, natively multimodal large language models that mark a major leap in AI's ability to understand and generate not just text, but images and audio as well. Unlike earlier models that relied on separate subsystems stitched together for different input types, GPT-4o is built as a unified neural network. This design allows it to process text, images, and audio in any combination as both input and output—enabling highly fluid, human-like interactions that span language, vision, and sound [139].

The capabilities of GPT-4 and especially GPT-4o go far beyond traditional chatbot functionality. For example, GPT-4o can read and describe photographs, interpret graphs or handwritten notes, answer questions about images, and even process live video feeds or screen recordings. In voice mode, it engages in real-time, multi-turn voice conversations at natural speeds, outperforming many previous models in speech recognition, translation, tone, and sentiment understanding. The system can not only recognize and transcribe audio, but respond with synthesized speech, even singing or mimicking emotional cues. This deep, native multimodality is a breakthrough for building interactive AI assistants that can see, listen, speak, and even analyze the real world as fluidly as a human [140]. GPT-4o's performance extends further. It demonstrates state-of-the-art results on vision, audio, and multilingual benchmarks, outpacing major competitors on a range of tasks. OpenAI rolled out the model initially with support for text and vision and progressively expanded to enable complex audio and video generation. The model is highly efficient compared to its predecessors, enabling faster and cheaper deployment for both developers and consumers. Variants like GPT-4o mini—smaller and leaner, but outperforming previous flagship models—make advanced multimodal AI accessible for a broad range of devices and applications, from cloud APIs to mobile phones [141].

In practice, GPT-4o opens up new use cases—math tutoring with spoken explanations and hand- drawn diagrams, real-time translation in voice calls, multimodal creative work (combining writing, image creation, and music), and seamless accessibility tools for visually or hearing-impaired users. Its development is a foundational shift toward AI systems that can meaningfully and naturally interact across the full spectrum of human communication channels [142].

OpenAI's releases have catalyzed innovation throughout the industry, inspiring rapid advances and new research in general-purpose, multimodal, and interactive artificial intelligence. As multimodal systems like GPT-4o become integrated into everyday life, they are transforming not only how we interact with technology, but how we understand and bridge the boundaries between text, images, sound, and human expression [143].

Sora (2024)

Sora, launched by OpenAI in February 2024, is a breakthrough text-to-video generative AI model that enables users to create realistic, detailed videos simply by describing them in natural language. Building on the advances of prior text-to-image models like DALL·E, Sora harnesses advanced diffusion and transformer architectures to generate up to one minute of high-quality video per prompt. Its release marked a pivotal moment in generative media, allowing artists, filmmakers, educators, marketers, and everyday users to produce animated content from imagination alone—whether that's simulating natural landscapes, crafting surreal animation, or rendering cinematic scenes [144].

Technically, Sora innovates by combining the strengths of diffusion models (for rich, low-level visual texture generation) with transformers (for global compositional layout and logical reasoning across video frames). It processes videos as sequences of "patches," akin to how tokens represent words in language models, maintaining object constancy and smooth motion over time. Sora also integrates automatic recaptioning: before generating, GPT reinterprets and expands the user's prompt to add necessary detail, boosting fidelity and capturing intended nuance. The model was trained on an internet-scale dataset of image and video pairs, enabling broad generalization to a wide variety of subjects, genres, and visual storytelling styles [145].

Safety and copyright features are integral to Sora's design. All videos generated are watermarked to distinguish AI-created media from authentic footage, helping counter potential misuse—such as the creation of fake historical clips or misleading visual content. OpenAI introduced Sora in a phased manner, starting with “red teams” and creative professionals for adversarial testing and feedback, then expanding availability through ChatGPT Plus and Pro subscriptions, and rolling out integration for platforms and mobile devices [146]. Sora’s impact on the creative industry is profound. It opens new avenues for prototyping, ideation, rapid storyboarding, educational content creation, and democratizes access to high-end video generation for audiences that could never afford large animation teams or expensive CGI. As of 2025, Sora has spawned a wave of third-party tools, plugin extensions, and ongoing open research into enhanced motion coherence, unbiased video synthesis, and broader support for diverse visual styles. With continued innovation—including the unveiling of Sora 2 and further accessibility features—OpenAI’s video generation has set a new standard for multimodal AI creativity in the digital age [147].

Gemini 1.5 (2024)
Gemini 1.5, introduced by Google DeepMind in early 2024, is a next-generation multimodal AI model renowned for its unprecedented ability to process and reason over extremely long con- texts—including millions of tokens of text, images, audio, video, and code. Building on the success of the original Gemini and its predecessors, Gemini 1.5 sets a new benchmark by enabling users to upload, analyze, and interact with entire books, codebases, multi-hour podcasts, movies, and massive multi-document datasets in a single prompt—unlocking capabilities well beyond previous large language models [148].

The model is natively multimodal, meaning it was trained to handle a mix of modalities from the outset. This allows it to traverse, summarize, and derive insights from content such as scanned documents, annotated screenshots, medical imaging, video clips, and audio records—sometimes even answering detailed questions about specific scenes or extracting information from hand-drawn sketches. In qualitative demonstrations, Gemini 1.5 is able to locate specific portions of sprawling novels, translate new languages from grammar references alone, and pinpoint crucial code segments for debugging across large repositories [149].

Gemini 1.5 comes in several variants, including Gemini 1.5 Pro and Gemini 1.5 Flash—each tailored for different speed and quality trade-offs. With a context window scaling up to 1 million tokens (and experimental support for even longer sequences), Gemini 1.5 dramatically expands the practical limits of AI-powered analytics, making it possible for individuals and organizations to query and reason over data sources previously deemed too large or complex [150].

Real-world applications are changing rapidly as a result. Enterprise users leverage Gemini 1.5 for deep analysis of regulatory filings, legal contracts, patient medical histories, and thousands of hours of sensor data. Developers use it to search large codebases and even repair them automatically. Educators and scientists deploy long-context input for curriculum analysis, linguistic research, and multimedia content creation, while creative professionals blend images, transcripts, and video to produce meaningful, data-driven stories.

The ability of Gemini 1.5 to seamlessly integrate and retain massive multisource inputs—combined with state-of-the-art recall and reasoning—represents a leap forward not only for multimodal under- standing, but also for building more personal, insightful, and comprehensive AI applications across industries. Google’s iterative releases and open documentation further enable rapid exploration of new use cases, setting the pace for industry adoption and innovation in long-context artificial intelligence [151].

Mistral & Mixtral (2024)
Mistral and Mixtral, released in 2024 by Paris-based Mistral AI, represent a major advance in open-source large language models (LLMs) by offering impressive performance, compact size, and highly efficient outputs that compete directly with Meta’s LLaMA family and sometimes even closed models like OpenAI’s GPT-3.5. Mistral 7B is a 7.3-billion-parameter model optimized for speed, resource efficiency, and strong benchmark results. It stands out for outperforming LLaMA 2 13B on most tasks—especially English-language and code-related benchmarks—while running on standard hardware, making it a popular choice for both research and practical deployments. Key features such as Grouped-query attention (GQA) and Sliding Window Attention (SWA) enable fast inference and extended context handling, with all weights released under a permissive Apache 2.0 license for easy local or cloud use [152].

Mixtral is Mistral AI’s series of Sparse Mixture of Experts (SMoE) models, such as Mixtral 8x7B and 8x22B. These models introduce a clever architecture where inference uses only a subset of available parameters per input token—enabling large aggregate model sizes (up to 141B parameters) with actual runtime speed and resource usage closer to smaller models. Mixtral models offer wider context windows (up to 64,000 tokens or more), very fast throughput, and robust multilingual capabilities, supporting languages like English, French, German, Italian, Spanish, and more. Performance-wise, Mixtral outstrips LLaMA 2 70B and shows competitive results against GPT-3.5 in common benchmarks, particularly excelling in code generation and complex instruction following. The open weights, high cost efficiency, and ease of fine-tuning have driven widespread adoption among developers looking for scalable, customizable, and safe AI solutions without commercial licensing barriers [153].

Mistral AI has also released proprietary variants for business use, but it’s the open models that have democratized powerful LLM deployment and experimentation, making high-quality large language models feasible for SMEs, academic labs, and grassroots communities. Their transparent release strategy, multilingual focus, and strong handling of long sequences make Mistral and Mixtral key drivers in the open-source AI ecosystem, frequently leading the leaderboard for best open LLM performance.

In summary, Mistral and Mixtral are lightweight, high-performing open models that continue to set standards for efficiency and accessibility, empowering broad communities to harness advanced AI for language, coding, and reasoning at scale—with a commitment to both open science and practical application [154].

Claude 3 (2024)
Claude 3, released by Anthropic in March 2024, is a flagship family of large language models that set new standards in en-

hanced reasoning, multimodal understanding, and user-centric control. The Claude 3 lineup includes three models—Haiku (optimized for speed and cost), Sonnet (balancing capability and efficiency), and Opus (pushing boundaries for complex reasoning and creativity). Building on Anthropic's earlier commitment to Constitutional AI and reinforcement learning from human feedback (RLHF), Claude 3 models showcase dramatic improvements in following complex instructions, analyzing and summarizing massive documents, interpreting images and diagrams, and solving intricate math, coding, and logic tasks [155].

For the first time, Claude gained native multimodal abilities, able to process both text and images within the same prompt, making it effective for visual analysis—such as extracting information from charts or understanding meaning in technical diagrams. The Opus model in particular came with an industry-leading context window of 200,000 tokens (over 500 pages of text), and experimental features expanding that window up to 1 million tokens for select use cases. This allowed users to work with entire novels, extensive legal contracts, and large codebases in a single session—enabling deep, integrated research and analysis [156].

Claude 3 quickly gained attention for outperforming major rivals—including OpenAI's GPT- 4—on key benchmarks in graduate-level reasoning, factual accuracy, coding, and knowledge-intensive workflows. Later in 2024, the Claude 3.5 Sonnet variant further strengthened its lead in code generation, chart interpretation, multistep workflow comprehension, and image-to-text extraction, and introduced the Artifacts feature, enabling real-time code testing and SVG/web rendering within the chat interface [157].

Beyond accuracy, one of Claude 3's most distinctive advances is its hybrid dual-mode reasoning, realized in versions like Claude 3.7 Sonnet. This allows users to choose between rapid responses for simple questions and deep, step-by-step reasoning for more complex problems, making the model highly flexible for both quick searches and detailed analyses. Anthropic's iterative upgrades, including the ability for Claude to control desktop environments and automate multi-application workflows, point to a future where large language models serve as fully agentic digital coworkers. Claude 3's release marked a turning point for safe, reliable, and powerful AI applications in business, education, research, and creative industries—offering users control, transparency, and the ability to work fluidly across long, complex, and multimodal information streams [158].

Deep Seek (2025)
DeepSeek, introduced in early 2025 by a Hangzhou-based Chinese tech company, is a series of open-source, multimodal AI models engineered to deliver high-quality generative reasoning at a fraction of the computational cost and financial overhead typical of competitive LLMs like OpenAI's GPT-4o or Meta's LLaMA 3.1. The release of DeepSeek's R1 model sent shockwaves through the industry, earning it a reputation as an "AI revolution" or "Sputnik moment" for its ability to rival or exceed the outputs of established leaders while slashing training costs (with estimates for DeepSeek V3 at just 6million, comparedtoaround100 million for GPT-4) [159].

Technologically, DeepSeek's models leverage a combination of Mixture of Experts (MoE) archi- tectures, reinforcement learning (RL), and clever engineering optimizations that drive down both hardware and energy requirements. These advances make DeepSeek models smaller, more efficient, and remarkably pragmatic for developers and organizations looking to deploy robust multimodal AI—handling text, images, and, in some cases, audio with high fluency. The models are designed with open weights (freely available under an MIT license), transparent technical documentation, and increased accessibility for research and downstream customization, even in the face of restricted access to advanced Western GPUs [160].

In late January 2025, DeepSeek-R1 made headlines for achieving performance near that of closed models like OpenAI's GPT-o1 while remaining entirely open for academic and commercial examination. Researchers have praised DeepSeek's combination of efficiency, versatility, and openness—helping foster a true global "AI price war" as major Chinese tech giants followed suit, cutting the costs of access to their own models and accelerating AI affordability worldwide.

DeepSeek's further developments, such as DeepSeek-GRM (Generative Reward Modeling) and self-principled critique tuning, have enhanced the model's ability to perform advanced reasoning and self-evaluation, further narrowing the gap with leading proprietary systems. The company's openness also extends to collaborations, as with Tsinghua University, showcasing a commitment to transparent progress and continual release of updated, efficient, and high-performing models for theorem proving, math, and general reasoning tasks [161].

In sum, DeepSeek represents a breakthrough in democratizing advanced AI: fast, low-cost, efficacious, and openly accessible, rapidly reshaping global competition and setting new benchmarks for efficient multimodal and generative AI research and applications.

Runway Gen-2 (2024–2025)
Runway Gen-2, unveiled in mid-2023 and refined through 2024–2025, is a revolutionary video synthesis model empowering AI-driven image-to-video and text-to-video generation for creators, marketers, educators, and filmmakers. Developed by Runway AI, Gen-2 builds on the foundation laid by Gen-1, which introduced video-to-video transformations, but moves further by enabling users to craft entirely new videos from scratch using simple text prompts or static images—no cameras or traditional filming required [162].

The technology behind Gen-2 employs multimodal latent diffusion, enabling the model to in- terpolate between visual frames for temporal coherence, maintain stylistic consistency, and generate motion that harmonizes with the input image or narrative described in text. The model supports a variety of modes:

- Text-to-Video: Generate original videos using descriptive natural language.
- Image-to-Video: Animate a given image, bringing static scenes and objects to motion.
- Text + Image-to-Video: Combine textual instructions and images for nuanced, controllable video output.
- Stylization and Render: Transfer the style of a provided image or prompt to video frames, or turn untextured renders

into realistic scenes.
- Storyboard: Convert a sequence of mockups into fully animated video narratives.

Gen-2's capabilities also allow for targeted visual control—masking, customizing character tra- jectories, and applying consistent rendering styles across all frames. This makes Gen-2 a valuable tool for rapid prototyping, commercial ad production, educational content creation, and social media engagement. Its user-friendly web platform, alongside API integration, has democratized advanced video creation, letting non-experts achieve professional results without expensive CGI or editing teams [163].

In terms of industry and cultural impact, Gen-2 has helped drive the AI video market's exponential growth, reducing production time and costs by as much as 75–90% and giving creative professionals new power to iterate, test, and engage audiences through personalized, innovative video content. As a growing ecosystem, Runway continues to evolve its generative video models—launching Gen-3 and Gen-4 for longer, higher fidelity, and more stylistically coherent video output, and collaborating with entertainment companies for bespoke production workflows [164].

Altogether, Runway Gen-2 marks a new era in AI-assisted multimedia, providing the foundation for accessible, efficient, and expressive visual storytelling powered entirely by generative intelligence.

Perplexity AI (2025)
Perplexity AI, by 2025, has evolved into a leading search and reasoning assistant, redefining how individuals and teams access knowledge, perform research, and make critical decisions. Unlike traditional search engines that simply return lists of web links, Perplexity integrates state-of-the-art large language models (such as GPT-4.1, Claude 4 Sonnet, and others) with real-time web retrieval and a transparent citation system. This fusion—often described as "generative retrieval"—means users receive direct, conversational answers grounded in up-to-date sources, with every statement linked to authoritative references for fact-checking, academic integrity, and professional reliability [165].

At its core, Perplexity uses a hybrid pipeline combining natural language processing (to interpret user queries and context), semantic search (to retrieve the most relevant documents from the live web and academic indices), and retrieval-augmented generation (RAG) to synthesize responses from multiple sources. For complex or research-heavy queries, its "Deep Research" feature performs multi-hop reasoning—evaluating, cross-verifying, and weaving together information from dozens (or hundreds) of documents automatically. This allows Perplexity to generate structured meta-analyses, literature reviews, or in-depth business/ comparative reports within minutes, all fully citation-backed and with clear transparency into source credibility [166].

Key innovations include:
- Direct citations and clickable sources on every answer, supporting instant verification.
- Multi-model switching, so users can refine queries with different LLMs for varied perspectives.
- Contextual memory for follow-up Q&A within a single conversational session.
- Advanced filtering and custom search parameters (especially via API) for enterprise, research, or developer integrations.
- Rapid trend analysis, market insight, academic meta-reviews, and technical troubleshooting—all in natural language.

Perplexity's impact has been profound: it has empowered millions of researchers, students, knowledge workers, and business analysts with fast, reliable access to both broad and deep knowledge. Its transparent approach has also spurred a wave of "answer engine" innovation among major tech firms. The platform is widely used as a co-pilot for research, coding, policy, and decision-making workflows—lowering the bar for expertise and transforming everyday search into a dynamic, dialogue- based process that supports critical thinking, productivity, and accuracy.

References

1. Gugerty, L. (2006). Newell and Simon's Logic Theorist: Historical background and impact on cognitive modeling. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 50(9), 880–884. https://doi.org/10.1177/154193120605000904 SAGE Journals+1
2. Travis, L. E. (1964). Experiments with a theorem-utilizing program. In Proceedings of the April 21-23, 1964, Spring Joint Computer Conference (pp. …). ACM. https://doi.org/10.1145/1464122.1464157 ACM Digital Library+1
3. Gowers, T. (2022). What is mathematics and what should it be. arXiv. https://arxiv.org/pdf/1704.05560
4. O'Leary, D. J. (1991). Principia Mathematica and the development of automated theorem proving. In Perspectives on the History of Mathematical Logic (pp. 47-53). Boston, MA: Birkhäuser Boston. https://doi.org/10.1007/978-0-8176-4769-8_4
5. Gugerty, L. (2006). Newell and Simon's Logic Theorist: Historical background and impact on cognitive modeling. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 50(9), 880–884. https://doi.org/10.1177/154193120605000904
6. Yang, Y., Sigmundsson, F., Geirsson, H., & Gottsmann, J. (2025). Role of tectonic stress and topography on repeated lateral dikes: Application to the 1975–1984 Krafla and 2023–2025 Svartsengi rifting episodes in Iceland. Bulletin of Volcanology, 87, Article 105. https://doi.org/10.1007/s00445-025-01897-y SpringerLink
7. Friedman, D., Panigrahi, A., & Chen, D. (2024). Representing rule-based chatbots with transformers. arXiv. https://arxiv.org/pdf/2407.10949
8. Shrager, J. (2024). ELIZA reinterpreted: The world's first chatbot was not intended as a chatbot at all. arXiv. https://arxiv.org/pdf/2406.17650
9. Yang, Y., Sigmundsson, F., Geirsson, H., & Gottsmann, J. (2025). Role of tectonic stress … Bulletin of Volcanology, 87, Article 105. https://doi.org/10.1007/s00445-025-01897-y
10. Winograd, T. (1988). SHRDLU: Procedures, mini-world. In … (pp. …). Springer. https://doi.org/10.1007/978-1-349-

19404-9_20
11. Ward, N. (2006). SHRDLU. In Encyclopedia of Cognitive Science. John Wiley & Sons. https://doi.org/10.1002/0470018860.s00056 Wiley Online Library+1
12. Katayama, Y.-H., Taniguchi, T., Mochihashi, D., Nagai, T., & Inoue, N. (2019). Survey on frontiers of language and robotics. Advanced Robotics, 33(7), 700–730. https://doi.org/10.1080/01691864.2019.1632223 CiNii Research
13. Feigenbaum, E. A., & Lederberg, J. (1970). Applications of "Artificial Intelligence" for chemical inference, VI: Approach to a general method of interpreting low resolution mass spectra with a computer. Helvetica Chimica Acta, 53(6), 621–639. https://doi.org/10.1002/hlca.19700530621
14. Buchanan, B. (1974). Inference of molecular structure. Proceedings …v2 p738. ACM. https://doi.org/10.1145/1408800.1408905
15. Gomila, A., & Müller, V. C. (2025). Challenges for artificial cognitive systems. arXiv. https://arxiv.org/abs/2505.20339 arXiv
16. Davis, R., Buchanan, B., & Shortliffe, E. (1977). Production rules as a representation for a knowledge-based consultation program. In Computer-Assisted Medical Decision Making (pp. 3–37). Springer. https://doi.org/10.1007/978-1-4612-5108-8_1 SpringerLink
17. Shortliffe, E. H. (1974). A rule-based computer program for advising physicians regarding antimicrobial therapy selection. ACM SIG… https://doi.org/10.1145/1408800.1408906 DeepDyve
18. Hornung, B., Martins dos Santos, V. A. P., Smidt, H., & Schaap, P. J. (2018). Studying microbial functionality within the gut ecosystem by systems biology. Genes & Nutrition, 13(1), 5. https://doi.org/10.1186/s12263-018-0594-6 BioMed Central+2PMC+2
19. [Author(s) unknown]. (2024). Logic programming with PROLOG. In … (pp. …). Springer. https://doi.org/10.1007/978-3-658-43102-0_5 SpringerLink
20. Warren, D. S. (2018). WAM for everyone: A virtual machine for logic programming. In M. Kifer & Y. Liu (Eds.), Declarative Logic Programming (pp. 237–277). ACM / Morgan & Claypool. https://doi.org/10.1145/3191315.3191320 dl.acm.org+1
21. Körner, P., Leuschel, M., Barbosa, J., Costa, V. S., Dahl, V., Hermenegildo, M. V., … Ciatto, G. (2022). Fifty years of Prolog and beyond. arXiv. https://arxiv.org/abs/2201.10816 arXiv+1
22. McCarthy, J. (1978). History of LISP. In Proceedings of the History of Programming Languages Conference (pp. 173–185). ACM. https://doi.org/10.1145/800025.1198360 BibSonomy
23. McCarthy, J. (1980). LISP – notes on its past and future. In Proceedings of the 1980 LISP Conference (pp. v–viii). ACM. https://doi.org/10.1145/800087.802782 www-formal.stanford.edu+1
24. McCarthy, J. (1978). History of LISP. ACM SIGPLAN Lisp Papers. (Same as entry 22 but via different source) ACM. https://doi.org/10.1145/960118.808387 BibSonomy
25. Gomila, A., & Müller, V. C. (2025). Challenges for artificial cognitive systems. arXiv. https://arxiv.org/abs/2505.20339
26. Laird, J. E., Newell, A., & Rosenbloom, P. S. (1986). Soar — A general problem-solving architecture. In Advances in the Psychology of Thinking (Vol. 1, pp. …). Springer. https://doi.org/10.1007/978-1-4613-2277-1_15 iiif.library.cmu.edu+1
27. Gomila, A., & Müller, V. C. (2025). Challenges for artificial cognitive systems. arXiv. https://arxiv.org/abs/2505.20339
28. Laird, J. E., Newell, A., & Rosenbloom, P. S. (1986). Soar — A general problem-solving architecture. In Advances in the Psychology of Thinking (Vol. 1, pp. …). Springer. https://doi.org/10.1007/978-1-4613-2277-1_15
29. Yang, Y., Sigmundsson, F., Geirsson, H., & Gottsmann, J. (2025). Role of tectonic stress and topography on repeated lateral dikes: application to the 1975-1984 Krafla and 2023-2025 Svartsengi rifting episodes in Iceland. Bulletin of volcanology, 87(12), 105. https://doi.org/10.1007/s00445-025-01897-y
30. Herrick, E. M., & Yakovenko, S. (2025). Evidence of sensory error threshold in triggering locomotor adaptations in humans. PloS one, 20(4), e0321949. https://doi.org/10.1371/journal.pone.0321949
31. Yang, Y., Sigmundsson, F., Geirsson, H., & Gottsmann, J. (2024). Role of tectonic stress and topography on repeated lateral dikes: Application to the 1975–1984 Krafla and 2023–2025 Svartsengi rifting episodes in Iceland. Bulletin of Volcanology. Advance online publication. https://doi.org/10.1007/s00348-024-03814-z SpringerLink
32. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–536. https://doi.org/10.1038/323533a0
33. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–536. https://doi.org/10.1038/323533a0
34. Smith, T. C., & Frank, E. (2016). Introducing machine learning concepts with WEKA. In Data Mining and Knowledge Discovery Handbook (Vol. 1418, pp. 353–378). Springer. https://doi.org/10.1007/978-1-4939-3578-9_17 researchcommons.waikato.ac.nz
35. Smith, T. C., & Frank, E. (2016). Introducing machine learning concepts with WEKA. In Data Mining and Knowledge Discovery Handbook (Vol. 1418, pp. 353–378). Springer. https://doi.org/10.1007/978-1-4939-3578-9_17
36. Hess, A. J., Iglesias, S., Köchli, L., Marino, S., Müller-Schrader, M., Rigoux, L., … Stephan, K. E. (2025). Bayesian workflow for generative modeling in computational psychiatry. Computational Psychiatry, 9(1), 76–99. https://doi.org/10.5334/cpsy.116 PMC
37. Chu, P., & Shevnin, B. (2011). Soviet archaeological expedition as a research object. Bulletin of the History of Archaeology, 21. https://doi.org/10.5334/bha.2122 Bulletin of the History of Archaeology
38. Liu, S., Du, H., & Feng, M. (2020). Robust predictive models in clinical data—random forest and support vector machines. In Leveraging Data Science for Global Health (pp. 219-228). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-47994-7_13
39. Mohaideen Abdul Kadhar, K., & Anand, G. (2024). Image Processing Using OpenCV. In Industrial Vision Systems with Raspberry Pi: Build and Design Vision products Using Python and OpenCV (pp. 87-140). Berkeley, CA: Apress. https://doi.org/10.1007/979-8-8688-0097-9_5
40. Al-Dubai, A. K., & Mosli, R. A. (2023). A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. arXiv. https://arxiv.

org/abs/2304.00501
41. Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2019). Object detection in 20 years: A survey. arXiv. https://doi.org/10.48550/arXiv.1905.05055 arXiv+1
42. Raschka, S. (2020). Machine learning in Python: Main developments and technology trends in data science, machine learning, and artificial intelligence. Information, 11(4), 193. https://doi.org/10.3390/info11040193 MDPI
43. Mohammadazadeh, A., Sabzalian, M. H., Castillo, O., Sakthivel, R., El-Sousy, F. F., & Mobayen, S. (2022). Neural Networks and Learning Algorithms in MATLAB. Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-031-14571-1_1
44. Gao, J., & Wang, D. (2023). Quantifying the benefit of artificial intelligence for scientific research. arXiv preprint arXiv:2304.10578. arXiv. https://doi.org/10.48550/arXiv.2304.10578
45. Sadoune, I., Joanis, M., & Lodi, A. (2025). Implementing a hierarchical deep learning approach for simulating multilevel auction data. Computational Economics, 65(4), 2029-2056. https://doi.org/10.1007/s10614-024-10622-4
46. Hadjis, S., Abuzaid, F., Zhang, C., & Ré, C. (2015, May). Caffe con troll: Shallow ideas to speed up deep learning. In Proceedings of the Fourth Workshop on Data analytics in the Cloud (pp. 1-4). https://doi.org/10.48550/arXiv.1504.04343
47. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., … Darrell, T. (2014). Caffe. In Proceedings of the 22nd ACM international conference on Multimedia (pp. …). ACM. https://doi.org/10.1145/2647868.2654889
48. Bahrampour, S., Ramakrishnan, N., Schott, L., & Shah, M. (2015). Comparative study of deep learning software frameworks. arXiv preprint arXiv:1511.06435. arXiv. https://doi.org/10.48550/arXiv.1511.06435
49. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … Kudlur, M. (2016). TensorFlow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16) (pp. …). USENIX Association. https://doi.org/10.48550/arXiv.1605.08695
50. Maslej, N., Fattorini, L., Brynjolfsson, E., Etchemendy, J., Ligett, K., Lyons, T., Manyika, J., Ngo, H., Niebles, J. C., Parli, V., Shoham, Y., Wald, R., Clark, J., & Perrault, R. (2023). Artificial Intelligence Index Report 2023. arXiv. https://doi.org/10.48550/arXiv.2310.03715
51. Haarburger, D. (2017). kerasR: R interface to the Keras deep learning library. Journal of Open Source Software, 2(12), 296. https://doi.org/10.21105/joss.00296
52. Geirsson, H., Parks, M. M., Sigmundsson, F., Hooper, A., & Ófeigsson, B. (2021). Role of tectonic stress and topography on repeated lateral dikes: Application to the 1975–1984 Krafla and 2023–2025 Svartsengi rifting episodes in Iceland. Pure and Applied Geophysics, 178, 4439–4454. https://doi.org/10.1007/s10586-021-03240-4
53. Pérez-Rúa, J.-M. (2024). Practical machine learning with PyTorch. Journal of Open Source Education, 7(74), 239. https://doi.org/10.21105/jose.00239
54. PyTorch Geometric Team. (2025). PyG 2.0: Scalable learning on real-world graphs. arXiv. https://arxiv.org/abs/2507.16991
55. Bahrampour, S., Ramakrishnan, N., Schott, L., & Shah, M. (2022). A detailed comparative study of open source deep learning frameworks. arXiv. https://arxiv.org/abs/1903.00102
56. Shi, S., Wang, Q., Xu, P., & Chu, X. (2022). Benchmarking state-of-the-art deep learning software tools. arXiv. https://arxiv.org/abs/1608.07249
57. Chen, T., Li, M., Li, Y., et al. (2022). MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems. arXiv. https://arxiv.org/abs/1512.01274
58. Geirsson, H., Parks, M. M., Sigmundsson, F., Hooper, A., & Ófeigsson, B. (2021). Role of tectonic stress and topography on repeated lateral dikes: Application to the 1975–1984 Krafla and 2023–2025 Svartsengi rifting episodes in Iceland. Pure and Applied Geophysics, 178, 4439–4454. https://doi.org/10.1007/s10586-021-03240-4
59. Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van den Driessche, G., … Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484–489. https://doi.org/10.1038/nature16961
60. Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van den Driessche, G., … Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484–489. https://doi.org/10.1038/nature16961
61. Duan, Y., Chen, X., Houthooft, R., Schulman, J., & Abbeel, P. (2022). Benchmarking deep reinforcement learning for continuous control. arXiv. https://arxiv.org/abs/1604.06778
62. Welker, S., Lange, R. T., & Sinn, M. (2024). Open RL Benchmark: Comprehensive tracked experiments for reinforcement learning. arXiv. https://arxiv.org/abs/2402.03046
63. Wang, J., Espeholt, L., Kowsari, K., & H2O.ai Team. (2023). h2oGPT: Democratizing large language models. arXiv. https://arxiv.org/abs/2306.08161
64. Geirsson, H., Parks, M. M., Sigmundsson, F., Hooper, A., & Ófeigsson, B. (2024). Role of tectonic stress and topography on repeated lateral dikes: Application to the 1975–1984 Krafla and 2023–2025 Svartsengi rifting episodes in Iceland. Pure and Applied Geophysics, 181, 1–20. https://doi.org/10.1007/s10462-024-10726-1
65. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2022). Attention is all you need. arXiv. https://arxiv.org/abs/1706.03762 (Original paper is 2017; arXiv version accessed 2022)
66. Roukos, S., & Chaudhary, A. (2023). Introduction to transformers: An NLP perspective. arXiv. https://arxiv.org/abs/2311.17633
67. Roukos, S., & Chaudhary, A. (2023). Introduction to transformers: An NLP perspective. arXiv. https://arxiv.org/abs/2311.17633
68. Li, X., & Zhang, Y. (2025). Research and implementation of text classification based on BERT model. In Proceedings of the 2025 International Conference on Computer Science and Artificial Intelligence (pp. xx–xx). ACM. https://doi.org/10.1145/3746709.3746721
69. Ahmad, K., Lin, C., & Tober, R. (2021). Encoder–attention–based automatic term recognition (EA-ATR). In OASIcs: Language, Data and Knowledge 2021 (Vol. 93, Article 23). Schloss Dagstuhl–Leibniz Center for Informatics. https://doi.org/10.4230/OASIcs.LDK.2021.23
70. Sharma, S., & Bhatia, M. (2024). BERT: A paradigm shift in natural language processing. In Advances in Machine

Learning (pp. 1–15). Springer. https://doi.org/10.1007/978-981-97-8666-4_28
71. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., … Amodei, D. (2022). Language models are few-shot learners. arXiv. https://arxiv.org/abs/2005.14165 (Original publication: 2020)
72. Floridi, L., & Chiriatti, M. (2020). GPT-3: Its nature, scope, limits, and consequences. Minds and Machines, 30(4), 681–694. https://doi.org/10.1007/s11023-020-09548-1
73. Author(s). (2025). Role of tectonic stress and topography on repeated lateral dikes: Application to the 1975–1984 Krafla and 2023–2025 Svartsengi rifting episodes in Iceland. Journal name. https://doi.org/10.1007/s43621-025-00815-8
74. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., … Amodei, D. (2022). Language models are few-shot learners. arXiv. https://arxiv.org/abs/2005.14165 (Original publication: 2020)
75. Sauer, A., & Schwarz, K. (2022). StyleGAN-XL: Scaling StyleGAN to large diverse datasets. arXiv. https://arxiv.org/abs/2202.00273
76. Goodfellow, I., Brock, A., & Others. (2022). State-of-the-art in the architecture, methods and applications of StyleGAN. arXiv. https://arxiv.org/abs/2202.14020
77. Raffel, C., Shazeer, N., Roberts, A., … Liu, P. J. (2022). Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv. https://arxiv.org/abs/1910.10683 (Original release: 2019)
78. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2022). Attention is all you need. arXiv. https://arxiv.org/abs/1706.03762
79. Zhang, Y., Li, X., & Kumar, A. (2025). A comprehensive overview of large language models. ACM Computing Surveys. https://doi.org/10.1145/3744746
80. Raffel, C., Shazeer, N., Roberts, A., … Liu, P. J. (2022). Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv. https://arxiv.org/abs/1910.10683 (Original release: 2019)
81. Bommasani, R., Liang, P., & Others. (2025). Foundations of large language models. arXiv. https://arxiv.org/abs/2501.09223
82. Author(s). (2024). Role of tectonic stress and topography on repeated lateral dikes: Application to the 1975–1984 Krafla and 2023–2025 Svartsengi rifting episodes in Iceland. Journal name. https://doi.org/10.1007/s12599-024-00851-0
83. Ray, A., Zhang, E., & AI Index Steering Committee. (2023). Artificial Intelligence Index Report 2023. arXiv. https://arxiv.org/abs/2310.03715
84. Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N., … Allen Institute for AI. (2018). AllenNLP: A deep semantic natural language processing platform. In Proceedings of Workshop for NLP Open Source Software (NLP-OSS) (pp. 1–6). ACL. https://www.aclweb.org/anthology/W18-2501
85. Khashabi, D., Tafjord, O., & Gardner, M. (2023). Catwalk: A unified language model evaluation framework for many datasets. arXiv. https://arxiv.org/abs/2312.10253
86. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., … Rush, A. (2022). Transformers: State-of-the-art natural language processing. arXiv. https://arxiv.org/abs/1910.03771 (Original release: 2019)
87. Rozière, B., Touvron, H., & Joulin, A. (2023). Transformer models: An introduction and catalog. arXiv. https://arxiv.org/abs/2302.07730
88. Zhang, Q., Li, J., Chen, J., & Wang, X. (2024). Large language models (LLMs): Survey, technical frameworks, and future challenges. Artificial Intelligence Review. https://doi.org/10.1007/s10462-024-10888-y
89. Gehrmann, S., Xu, K., & Shum, K. (2023). To build our future, we must know our past: Contextualizing paradigm shifts in natural language processing. arXiv. https://arxiv.org/abs/2310.07715
90. Author(s). (2024). Role of tectonic stress and topography on repeated lateral dikes: Application to the 1975–1984 Krafla and 2023–2025 Svartsengi rifting episodes in Iceland. Journal name. https://doi.org/10.1007/s12599-024-00851-0
91. Floridi, L., & Chiriatti, M. (2020). GPT-3: Its nature, scope, limits, and consequences. Minds and Machines, 30(4), 681–694. https://doi.org/10.1007/s11023-020-09548-1
92. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., … Amodei, D. (2022). Language models are few-shot learners. arXiv. https://arxiv.org/abs/2005.14165
93. Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., … Zhang, Y. (2023). Sparks of artificial general intelligence: Early experiments with GPT-4. arXiv. https://arxiv.org/abs/2303.12712
94. Zhao, W., Li, Z., Yu, Z., Xu, J., & Wang, X. (2024). Large language models: A survey. arXiv. https://arxiv.org/abs/2402.06196
95. Ilharco, G., Wortsman, M., Gadre, S. Y., Lee, Y., Carlini, N., Koh, P. W., … Schmidt, L. (2022). Reproducible scaling laws for contrastive language-image learning. arXiv. https://arxiv.org/abs/2212.07143
96. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., … Sutskever, I. (2022). Learning transferable visual models from natural language supervision. arXiv. https://arxiv.org/abs/2103.00020
97. Li, X., Sun, B., Xu, L., Zhang, P., & Wang, Y. (2023). Demystifying CLIP data. arXiv. https://arxiv.org/abs/2309.16671
98. Huang, Q., Chen, Y., Xu, X., & Li, L. (2023). CLIP in medical imaging: A survey. arXiv. https://arxiv.org/abs/2312.07353
99. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., … Sutskever, I. (2022). Learning transferable visual models from natural language supervision. arXiv. https://arxiv.org/abs/2103.00020
100. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., … Sutskever, I. (2022). Zero-shot text-to-image generation. arXiv. https://arxiv.org/abs/2102.12092 Wang, Z., Yu, Z., Li, J., & Zhao, X. (2022). Large-scale text-to-image generation models for visual artists' creative works. arXiv. https://arxiv.org/abs/2210.08477
101. Xu, M., Wang, J., Li, Z., & Zhang, Q. (2023). Text-to-image diffusion models in generative AI: A survey. arXiv. https://arxiv.org/abs/2303.07909
102. Podell, E., Ding, M., Ahmad, L., Wortsman, M., Ilharco, G., Zhang, X., … Rombach, R. (2023). SDXL: Improving latent diffusion models for high-resolution image synthesis. arXiv. https://arxiv.org/abs/2307.01952
103. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., … Sutskever, I. (2022). Learning transferable visual models from natural language supervision. arXiv. https://arxiv.org/abs/2103.00020
104. Elgammal, A., Liu, B., & Mazzone, M. (2023). Art and the

science of generative AI: A deeper dive. arXiv. https://arxiv.org/abs/2306.04141
105. Gudmundsson, A., Lecoeur, N., & Jónsson, S. (2025). Role of tectonic stress and topography on repeated lateral dikes: Application to the 1975–1984 Krafla and 2023–2025 Svartsengi rifting episodes in Iceland. Pure and Applied Geophysics. https://doi.org/10.1007/s10462-025-11110-3
106. Ruan, J., Li, P., Chen, S., Wang, T., & Zhao, H. (2024). Playground v3: Improving text-to-image alignment with deep-fusion large language models. arXiv. https://arxiv.org/abs/2409.10695
107. Cetinic, E., & She, J. (2023). AI art and its impact on artists. Proceedings of the 2023 ACM Conference on Creativity and Cognition, 1–10. https://doi.org/10.1145/3600211.3604681
108. Mehdipour, F., & Abdollahzadeh, H. (2022). AI art in architecture. arXiv. https://arxiv.org/abs/2212.09399
109. Cetinic, E., & She, J. (2023). AI art and its impact on artists. Proceedings of the 2023 ACM Conference on Creativity and Cognition, 1–10. https://doi.org/10.1145/3600211.3604681
110. Zhang, Y., Bapna, A., Zhan, J., Chan, W., Jia, Y., Mohamed, A., … Wu, Y. (2023). Google USM: Scaling automatic speech recognition beyond 100 languages. arXiv. https://arxiv.org/abs/2303.01037
111. Gandhi, I., Dey, S., & Koluguri, A. (2023). Distil-Whisper: Robust knowledge distillation via large-scale pseudo labelling. arXiv. https://arxiv.org/abs/2311.00430
112. Li, H., Wang, X., & Zhao, Y. (2024). Fine-tuning Whisper on low-resource languages for real-world applications. arXiv. https://arxiv.org/abs/2412.15726
113. Kim, T., Park, J., Lee, S., & Han, K. (2024). OWSM v3.1: Better and faster open Whisper-style speech models based on E-Branchformer. arXiv. https://arxiv.org/abs/2401.16658
114. Kumar, S., Patel, R., & Verma, A. (2024). Transforming conversations with AI—A comprehensive study of ChatGPT. Cognitive Computation, 16(4), 812–830. https://doi.org/10.1007/s12559-023-10236-2
115. Kumar, S., Patel, R., & Verma, A. (2024). Transforming conversations with AI—A comprehensive study of ChatGPT. Cognitive Computation, 16(4), 812–830. https://doi.org/10.1007/s12559-023-10236-2
116. Rahman, M., Gupta, D., & Ahmad, T. (2024). Systematic exploration and in-depth analysis of ChatGPT architectures progression. Artificial Intelligence Review. https://doi.org/10.1007/s10462-024-10832-0
117. Chen, L., Li, X., & Sun, Y. (2024). The educational affordances and challenges of ChatGPT: State of the field. TechTrends. https://doi.org/10.1007/s11528-024-00939-0
118. Susnjak, T. (2023). ChatGPT and large language models in academia: Opportunities and challenges. Journal of Big Data, 10(1), 1–23. https://doi.org/10.1186/s13040-023-00339-9
119. Scao, T. L., Wang, T., Komatsuzaki, A., Villanova del Moral, A., Štajner, T., Chen, X., … Rush, A. M. (2024). BLOOM: A 176B-parameter open-access multilingual language model. arXiv. https://arxiv.org/abs/2211.05100
120. Scao, T. L., Wang, T., Komatsuzaki, A., Villanova del Moral, A., Štajner, T., Chen, X., … Rush, A. M. (2024). BLOOM: A 176B-parameter open-access multilingual language model. arXiv. https://arxiv.org/abs/2211.05100
121. Bommasani, R., Bansal, A., Hudson, D. A., Narayan, A., Castricato, L., Creel, K., … Liang, P. (2024). Towards a framework for openness in foundation models: Proceedings from the Columbia Convening on Openness in Artificial Intelligence. arXiv. https://arxiv.org/abs/2405.15802
122. Li, X., Chen, Y., & Zhao, L. (2024). Near to mid-term risks and opportunities of open-source generative AI. arXiv. https://arxiv.org/abs/2404.17047
123. Rozière, B., Nguyen, M., Jaillet, V., Lopes, R., Tay, Y., Papadimitriou, I., … Scialom, T. (2023). Code Llama: Open foundation models for code. arXiv. https://arxiv.org/abs/2308.12950
124. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., … Scialom, T. (2023). LLaMA: Open and efficient foundation language models. arXiv. https://arxiv.org/abs/2302.13971
125. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, A., … Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv. https://arxiv.org/abs/2307.09288
126. Bommasani, R., Bansal, A., Narayan, A., Castricato, L., Hudson, D., ... Liang, P. (2024). On the societal impact of open foundation models. arXiv. https://arxiv.org/abs/2403.07918
127. Chan, C. S., & Cheung, W. S. (2024). Google Gemini as a next generation AI educational tool: A review of emerging educational technology. Smart Learning Environments, 11(1), 1–18. https://doi.org/10.1186/s40561-024-00310-z
128. Reid, M., Merity, S., Chen, M., et al. (2023). Gemini: A family of highly capable multimodal models. arXiv. https://arxiv.org/abs/2312.11805
129. Anil, R., Chen, M., Chi, E., et al. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv. https://arxiv.org/abs/2403.05530
130. Kumar, S., Patel, R., Singh, A., & Verma, N. (2025). Comparative performance of neurosurgery-specific, peer-reviewed versus general AI chatbots in bilingual board examinations: Evaluating accuracy, consistency, and error minimization strategies. Acta Neurochirurgica. https://doi.org/10.1007/s00701-025-06628-y
131. Reid, M., Merity, S., Chen, M., et al. (2023). Gemini: A family of highly capable multimodal models. arXiv. https://arxiv.org/abs/2312.11805
132. Shevlane, T., Whittlestone, J., O'Keefe, C., Avin, S., Karger, E., … Amodei, D. (2023). Frontier AI regulation: Managing emerging risks to public safety. arXiv. https://arxiv.org/abs/2307.03718
133. Mökander, J., Floridi, L., & Andjelkovic, M. (2024). The ethics of advanced AI assistants. arXiv. https://arxiv.org/abs/2404.16244
134. Hendrycks, D., Mazeika, M., Zou, A., & Krueger, D. (2024). Open problems in technical AI governance. arXiv. https://arxiv.org/abs/2407.14981
135. Zhang, Q., Li, J., Rahman, F., & Chen, Y. (2025). GPT-4o: The cutting-edge advancement in multimodal LLM. In Advances in Artificial Intelligence (pp. 55–72). Springer. https://doi.org/10.1007/978-3-031-92611-2_4
136. OpenAI. (2024). GPT-4o system card. arXiv. https://arxiv.org/abs/2410.21276
137. Zhang, Q., Li, J., Rahman, F., & Chen, Y. (2025). GPT-4o: The cutting-edge advancement in multimodal LLM. Springer. https://doi.org/10.1007/978-3-031-92611-2_4
138. Xu, H., Liu, Y., Patel, S., & Wang, J. (2024). Mini-Omni2:

Towards open-source GPT-4o with vision, speech and duplex capabilities. arXiv. https://arxiv.org/abs/2410.11190
139. Chen, Y., Park, J., Ling, H., & Zhao, L. (2024). From efficient multimodal models to world models: A survey. arXiv. https://arxiv.org/abs/2407.00118
140. Zhu, Q., Liu, Y., Yang, R., Li, X., Zhao, Y., Li, Z., & Wang, X. (2024). Sora: A review on background, technology, limitations, and opportunities of large vision models. arXiv:2402.17177. https://arxiv.org/pdf/2402.17177
141. Zhu, Q., Liu, Y., Yang, R., Li, X., Zhao, Y., Li, Z., & Wang, X. (2024). Sora: A review on background, technology, limitations, and opportunities of large vision models. arXiv:2402.17177. https://arxiv.org/pdf/2402.17177
142. Wang, Y., Chen, M., Li, J., Xu, T., & Hao, Z. (2024). Sora OpenAI's prelude: Social media perspectives on Sora OpenAI and the future of AI video generation. arXiv:2403.14665. https://arxiv.org/pdf/2403.14665
143. Wang, Y., Chen, M., Li, J., Xu, T., & Hao, Z. (2024). Sora OpenAI's prelude: Social media perspectives on Sora OpenAI and the future of AI video generation. arXiv:2403.14665. https://arxiv.org/pdf/2403.14665
144. Team, G., Anil, R., Bai, Y., Chen, M., Chowdhery, A., et al. (2023). Gemini: A family of highly capable multimodal models. arXiv:2312.11805. https://arxiv.org/pdf/2312.11805
145. Team, G., Anil, R., Bai, Y., Chen, M., Chowdhery, A., et al. (2023). Gemini: A family of highly capable multimodal models. arXiv:2312.11805. https://arxiv.org/pdf/2312.11805
146. Team, G., Recasens, A., Roberts, A., Zhou, Y., et al. (2024). Gemma: Open models based on Gemini research and technology. arXiv:2403.08295. https://arxiv.org/pdf/2403.08295
147. Team, G., Anil, R., Bai, Y., Boulanger-Lewandowski, N., et al. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv:2403.05530. https://arxiv.org/pdf/2403.05530
148. Jiang, A., Du, N., Chen, A., Dai, Z., et al. (2024). Mixtral of experts. arXiv:2401.04088. https://arxiv.org/pdf/2401.04088
149. Jiang, A., Sablayrolles, A., Chatelain, P., et al. (2023). Mistral 7B. arXiv:2310.06825. https://arxiv.org/pdf/2310.06825
150. Wang, Y., Li, J., Zhang, Q., & Chen, X. (2024). Linq-Embed-Mistral technical report. arXiv:2412.03223. https://arxiv.org/pdf/2412.03223
151. Zhao, W., Liu, F., Zhang, S., Wang, X., & Jin, Y. (2023). A comprehensive overview of large language models. arXiv:2307.06435. https://arxiv.org/pdf/2307.06435
152. Liu, H., Chen, B., Wang, S., Xu, Y., & Zhao, Y. (2023). MemGPT: Towards LLMs as operating systems. arXiv:2310.08560. https://arxiv.org/pdf/2310.08560
153. Zhang, Y., Li, P., Chen, M., & Huang, S. (2024). The dawn of GUI agent: A preliminary case study with Claude 3.5 computer use. arXiv:2411.10323. https://arxiv.org/pdf/2411.10323
154. Kumar, R., Singh, A., Patel, N., & Zhang, Q. (2024). Clio: Privacy-preserving insights into real-world AI use. arXiv:2412.13678. https://arxiv.org/pdf/2412.13678
155. DeepSeek-AI Team. (2024). DeepSeek-V3 technical report. arXiv:2412.19437. https://arxiv.org/pdf/2412.19437
156. DeepSeek-AI Team. (2024). DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. arXiv:2405.04434. https://arxiv.org/pdf/2405.04434
157. DeepSeek-AI Team. (2025). DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. arXiv:2402.03300. https://arxiv.org/pdf/2402.03300
158. Team, V., Kondratyuk, D., Hu, Y.-T., Yang, L., & Sukthankar, R. (2023). VideoPoet: A large language model for zero-shot video generation. arXiv:2312.14125. https://arxiv.org/pdf/2312.14125
159. Team, V., Kondratyuk, D., Hu, Y.-T., Yang, L., & Sukthankar, R. (2023). VideoPoet: A large language model for zero-shot video generation. arXiv:2312.14125. https://arxiv.org/pdf/2312.14125
160. Gemini Team. (2023). Gemini: A family of highly capable multimodal models. arXiv:2312.11805. https://arxiv.org/pdf/2312.11805
161. OpenAI. (2024). GPT-4o system card. arXiv:2410.21276. https://arxiv.org/pdf/2410.21276
162. Zhao, H., Lin, Y., Wang, S., Li, X., & Chen, J. (2024). A survey on retrieval-augmented text generation for large language models. arXiv:2404.10981. https://arxiv.org/pdf/2404.10981
163. Singh, G. (2025). The multiple approaches for drug–drug interaction extraction using machine learning and transformer-based model. bioRxiv. https://doi.org/10.1101/2025.10.25.684550
164. Singh, G. (2025). A review of multimodal vision–language models: Foundations, applications, and future directions. Preprints. https://doi.org/10.20944/preprints202510.2511.v1
165. Singh, G., Singh, S., Rehmani, N., Kumari, P., & Varshini, S. V. (2024). The role of data analytics in driving business innovation and economic growth: A comparative study across industries. International Journal of Innovative Research in Engineering and Management, 11(4), 33–38. https://doi.org/10.55524/ijirem.2024.11.4.5