ICON Journal of Engineering Applications of Artificial Intelligence
Volume 02 | Issue 04 | 2026
e-ISSN: 3051-3820
p-ISSN: 3051-3812
Journal homepage: https://iconpublishers.com/icon-j-eng-app-ai/

Research Article

DOI: 10.5281/zenodo.19723096

A Historical and Systematic Review of Generative Artificial Intelligence: Rule-Based, Model-Based, Deep Generative

***Prof. (Dr.) Sumit Kumar Gupta [1], Dr. Rajani Vyas [2]**
[1] Professor, Department of Physics, St. Wilfred's PG College, Jaipur, Rajasthan, India.
[2] Associate Professor, Department of Botany, St. Wilfred's PG College, Jaipur, Rajasthan, India.

Corresponding author: Prof. (Dr.) Sumit Kumar Gupta
Professor, Department of Physics, St. Wilfred's PG College, Jaipur, Rajasthan, India.

Received Date: 10 March 2026

Published Date: 25 April 2026

Abstract
In recent years, generative artificial intelligence (GAI) has achieved significant success, enabling individuals to create texts, images, videos, and even computer codes, while providing insights that may not be accessible through traditional tools. To foster future research, this paper presents a concise overview of the current and historical trends of GAI over the past 70 years. The successes are categorized into four groups: (i) Generative systems can be rule-based, operating according to specific rules and instructions, or model- based, generating outputs by applying models to input data. (ii) Generative algorithms are those that create new content using statistical or graphical models. (iii) Deep generative methodologies employ deep neural networks to learn how to generate new content from data. (iv) Trained foundation models, which are trained on large datasets, can generate a variety of images. The paper also discusses successful applications of generative systems and identifies open challenges arising from unresolved issues. Furthermore, this paper outlines potential research areas to better realize and exploit GAI technologies.

Keywords: artificial intelligence, foundation model, generative method.

I. Introduction

Generative Artificial Intelligence (GAI) is a type of artificial intelligence algorithm and model that can generate novel content, such as text, images, videos, and problem- solving strategies, with the creativity and flexibility of humans. The last few years have seen GAI experience an unprecedented progress. It is worth noting that the AI application ChatGPT [1] is able to converse with humans in more than 80 languages and can execute a variety of tasks that require a text-based answer. It also has the ability to create visual, audio and multimedia content. This incredible advancement is due to the evolution of artificial intelligence that has taken over 50 years to achieve. Such milestones as Deep Learning, Transformer architecture, and foundation models have been established. This article is a historical analysis of GAI in terms of its systematic review. The area is modern GAI made possible by programmable computing systems. The development of GAI is divided into four big stages, which show the great technological progress.

(1) Rule-based generative systems: Early history Rule-based systems were first developed in the 1950s, and had the ability to produce content autonomously. Such systems were based on predetermined rules that were designed by human professionals. At this stage, the expert systems were successful in domain specific tasks but their scalability and flexibility was limited.

(2) Generative algorithms based on models: Generative algorithms using statistical and mathematical models were subsequently developed by researchers. This phase increased GAI in fields like machine learning, neural networks, computer graphics, and computer vision. Applications such as computer animation and procedural content generation turned out to be more dependable and started to take over manual human activities.

(3) Deep generative techniques: As the amount of computational resources and data increases, deep neural networks [2], [3] have been shown to perform better at content generation [4]. Generative models like autoregressive models [5] and diffusion models [6] became potent and influential methods of generating high- quality outputs. Also, the methods of deep learning greatly contributed to the development of computer graphics and visual synthesis [7], which allowed creating realistic and scalable content in various fields. Recent developments in deep learning have greatly enhanced the ability and upscaling of generative models in open environments. These models have now the ability to work in various fields, process unstructured data, and produce high-fidel results with minimal human involvement. Deep generative systems have been found to be applicable to the real world due to the integration of big data and high-performance computing, which has allowed strong generalization and flexibility.

(4) Foundation models:

A recent breakthrough in GAI is the emergence of generative pre-trained transformers (GPTs) [8]-[10], a well- known type of foundation model. Transformer-based models are models that operate on large datasets and use massive deep learning methods. Their parameter and training data scalability results in the unprecedented performance improvements such as the high-quality content generation, natural human-like interaction and task generalization. Subsequently, foundation models have emerged as the engine of the current content generation systems, such as ChatGPT. This rapid growth of GAI has changed content/services production so that it is now possible to create multimedia information and also more structured outputs which include plans, source code and even biological sequences. Implementation of the GAI technologies has been extended to various sectors especially after foundation models have been successful. Generative approaches have been used in traditional industries like manufacturing, in new arenas like autonomous driving [11] and in the highly technical science of molecular design [12]. The development of GAI technologies over time (Fig. 1) demonstrates several milestones, such as Generative Adversarial Networks (GANs), diffusion models (DDPM), Neural Radiance Fields (NeRF), and 3D Gaussian representations. These advancements underscore the accelerated rate of development of generative methods between theoretical framework and practical, scalable solutions.

This paper will discuss the exemplary generative methods in the following sections, discusses their advantages and disadvantages and examines successful examples in the wide variety of disciplines. In addition, it addresses open research issues and provides a description of the possible future directions of the sphere of generative artificial intelligence.

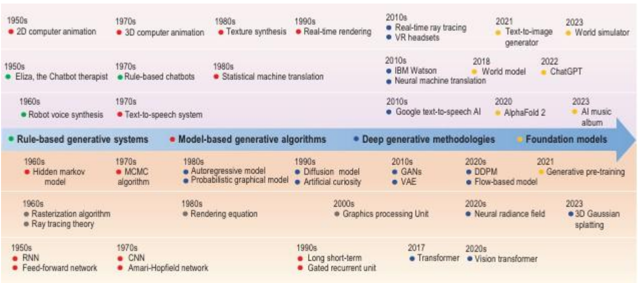

Figure_1: Timeline of the development of GAI methods and applications.

RULE-BASED GENERATIVE SYSTEMS

Automatic generation of data can be traced back to the 1950s when the symbolic artificial intelligence developed. In this initial stage, which is conceptually represented in Fig. 2(a), scientists created rule-based systems by formalizing human knowledge in computational systems. These systems were based on clearly defined logical rules to do generative tasks. Typically, such programs [13] consisted of two main components: a generation engine and an interpreter. The generation engine has the task of generating data based on specified logical and symbolic operations. It is constructed on a knowledge base which comprises of rules and facts. The pairs of rules that human experts develop include antecedents and consequents in the shape of condition-action pairs [14]. Such rules are symbolically coded and inbuilt into the system via programming. Nonetheless, these rules tend to be task-oriented and do not possess the ability to generalize. Therefore, subsequent works [15] extended systems based on rules to deal with other applications like dialogue systems, machine translation, and automated reasoning.

Besides rules, facts, which reflect structured information about propositions [13], properties [14] and relationships among entities [16] are also part of the knowledge base. The generation engine considers the rules one by one when an input signal that carries factual information is given. When a rule matching is found, the system creates the new facts by executing a matching rule. The generation cycle is complete as this process of reasoning is repeated until a set of predetermined termination conditions are met. Even though the first artificial intelligence was based on rule-based generative systems, such systems are limited by scalability and adaptability because they require hand-written rules. However, they still have value to learn about the historical development of generative artificial intelligence and the shift to more sophisticated learning-based methods.

Interpreter and Applications of Rule- Based Generative Systems

The interpreter of rule-based generative programs is an important element in enhancing transparency and explainability. It helps humans to reason the logic behind the functions carried out by the generation engine. In particular, the interpreter converts the underlying rules and actions to human readable explanations [17]. This is done by translating the logical form of rules into descriptive forms that will enable users to easily understand how decisions are arrived at. It is important to note that the interpreter itself can be executed through rule-based mechanisms, but these rules are different than the ones implemented in the process of generation. In consequence, generative systems that are based on rules are commonly described as white-box models, in which the reasoning process can be completely explained. This transparency helps in debugging, verification as well as changing the system even when the generation process gets complicated. Generative tasks Generative tasks were commonly used between the 1950s and 1990s using expert systems based on user-written rules and knowledge bases. Such systems worked well especially in areas where specialized knowledge was needed. Early chatbots, machine translation systems and speech synthesis systems are prominent applications. ELIZA, which was designed by Joseph Weizenbaum in 1966, is one of the first examples of a rule-based chatbot. ELIZA was a simulation of a psychotherapist that used pattern matching and response generation to respond to user inputs according to a set of rules. Its success was due to the fact that it was working within a limited conversational space, where rule-based methods could convincingly simulate human interaction. Later research [18] expanded the capabilities of chatbots to imitate other roles, including patients with psychological disorders. These systems however had no understanding of the context and were confined to tasks that were small in definition. It was also in this period that machine translation systems were proposed with the first being proposed in the late 1950s. Such systems featured elaborate linguistic and grammatical regulations, as well as the structured bodies of knowledge with language-specific information. Gradually, the rule-based translation systems were further advanced with the help of the contributions of computational linguists and domain experts. One remarkable instance is SYSTRAN [19], launched in 1968, that was still used during decades in the web-based translation services.

Likewise, the initial speech synthesis systems were designed in the rule-based way. The model suggested by Gunnar Fant and others in the 1960s [20] used linguistic and phonetic rules to recreate speech production and relate phonetic units. Though the speech synthesized had a mechanical sound and was not really fluent, it was understandable and could be used in practice, including assistive technologies and automatic announcements.

Rule-based generative systems have limitations, although the initial success. One of the biggest struggles is dealing with situations which are not covered by the prior rules. In practice, it is not viable to foresee and code all the possible scenarios, resulting in a lack of robustness and flexibility. Moreover, the complexity of the system correspondingly increases the amount of required rules, so system design, maintenance, and updates become more and more resource-consuming. These constraints eventually spurred a shift to data-driven and learning-based generative methods.

Generative Algorithm Model-Based

To overcome the limitations that are inherent in rule-based methods, researchers considered generative algorithms based on mathematical and statistical principles. This paradigm has become the default in generative artificial intelligence (GAI) as shown in Fig. 2(b) conceptually. Here we discuss model- based methods, especially methods by statistical machine learning and neural computation, as they are of considerable importance to contemporary GAI.

A. Statistical Machine Learning Models

Statistical machine learning is concerned with the development of algorithms that do not rely on a set of explicitly defined rules but learn patterns directly based on data. These models can be generally classified into discriminative and generative models [21]. Unlike discriminative models, which predict outputs based on the inputs provided, generative models learn the underlying data distribution and produce new samples by inferring or sampling. Generative modeling has grown to be explicit and implicit since the 1960s. Explicit modeling methods presume the characterization of the data distribution directly. The most notable of these is the probabilistic graphical models [22], in which nodes are random variables and edges are probabilistic relationships. Such models are based on generating data as an inference problem on partially observed variables. The likelihood-based methods, including the expectation–maximization (EM) algorithm, are commonly used to optimize these models [23].

An important assumption made in most probabilistic models is the Markov property [24] which assumes that future states are only dependent on the present state. Hidden Markov Model (HMM) [25] is a continuation of this idea, but it adds latent variables that control observable data. This framework has been widely used in sequential data generation. Markov random fields and Bayesian networks [29] have been further developed and are effective to model dependencies between various variables and generate structured data.

Another type of explicit generative models is autoregressive models [5], although they are more useful with sequential data. These models produce data elements sequentially, conditioned by already generated data elements. It was used early on in speech and language modeling in the 1980s [30], and then in neural autoregressive models in the 1990s [31].

More recently, scales of autoregressive models to large neural architectures have been developed, starting with the Transformer architecture [8], and are the basis of more recent neural systems, including GPT models [9]. Implicit generative modeling methods have become of interest as well. An example is the normalizing flows [32], which converts simple probability distributions into complex probability distributions by way of invertible mappings. Diffusion-based models are based on non-equilibrium thermodynamics and can be seen as a stochastic process model of data generation [6]. This was formalized subsequently in probabilistic models [33]–[35] and grouped together in the form of differential equations [36]. Never fully exploited, their successors, probabilistic diffusion models [37], [38] and flow- matching methods [39], took a central role in the current generative AI.

B. Generative Models Based on Neural Networks

Another major branch of model-based generative algorithms is based on artificial neural networks. Artificial neurons have their roots in early research of biological neural systems [40], and their mathematical basis in linear regression models [41], [42]. Neural networks can be used to generate data because they can model complex data distributions by using nonlinear activation functions. A number of neural network designs have been designed to facilitate generative modelling. Some of the earliest neural networks were the feed-forward neural networks [44], which are largely popular because of their simplicity. CNNs [45] are specifically efficient in image-related tasks whereas Recurrent Neural Network (RNNs) allow modelling the sequential data. RNNs have their theoretical basis in the models of statistical mechanics that were suggested by Wilhelm Lenz [47] and Ernst Ising [48].

The more advanced architecture like the Amari-Hopfield network [50], [51] was capable of associative memory, which enabled storage and retrieval of patterns. Later, Long Short-Term Memory (LSTM) networks [52] enabled the capability to capture long-range connections between sequential data, greatly boosting text and speech generation.

Neural generative modeling is further expanded into energy-based models. These are models that take the form of an energy function on data configurations, with lower energy implying higher probability. One well-known such model is the Restricted Boltzmann Machine (RBM) [55], which learns hierarchical data representations with a two-layer architecture.

C. Generative Model training and Inference

Modern neural generative models heavily depend on the backpropagation algorithm. Backpropagation, first described by Seppo Linnainmaa [56] and popularised by Paul Werbos [57], is an efficient approach to computing gradients based on the chain rule. It has been the commonly used technique to optimize neural networks since the 1980s [58], [59].

Generative models are used to make inferences by generating new data samples based on the learned distributions. Nevertheless, precise inference is usually computationally infeasible [60], thus giving way to approximate inference methods. There are two main methods that are commonly employed: Stochastic approximation: Markov Chain Monte Carlo (MCMC) [61], [62] are methods that build up Markov chains whose stationary distribution is an approximation to the target data distribution. Variational inference: This method estimates complicated distributions with the help of less complicated parameterized distributions and optimizes them with metrics like the evidence lower bound (ELBO) [67], [68].

D. Discussion and Limitations

Model-based generative algorithms have much better generalization and flexibility than rule-based systems. They have allowed significant improvements, including the development of statistical machine translation that overtook rule- based systems in the 1980s, and came to dominate speech and language processing in the 1990s. Also, these methods have been used in the visual data generation tasks e.g. texture synthesis and image fusion.

However, challenges remain. High dimensionality data presents computational complexity, commonly known as the curse of dimensionality and constrains scalability in some real world applications. Nevertheless, the model- based approaches provided the fundamental basis of the modern deep generative models and are still relevant in the evolution of advanced GAI systems.

Graphics-Based Generative Models

Generative approaches to graphics concentrate on visual representation using physically based modeling and rendering. These methods are based on the theory of vision developed by David Marr [69] that offers a computational structure to reconstitute the shape and appearance of real world scenes. In this paradigm, the process of rendering, which involves combining materials, textures, lighting and geometry to generate realistic visual outputs, is used to generate content. More recent work [7], [70] has incorporated deep learning methods into graphics pipelines, thus applying graphics-based methods to the field of generative artificial intelligence (GAI). Graphics-based methods are used to build three- dimensional (3D) scenes with explicit or implicit representations. Geometric primitives, point clouds [71] and voxels [72] are intuitive and interpretable, and have been popular in early work. Conversely, implicit representations represent scene information using deep neural networks, and the continuous functions can be rendered at any resolution and viewpoint.

The graphics-based model rendering techniques can be divided into two broad categories: rasterization and ray tracing. Computationally efficient Rasterization methods [73] – [75] are a good fit where a real-time rendering is required, e.g. in video games and interactive simulations. In contrast, more realistic images (with reflections, refractions, shadows) are generated using the ray tracing methods [76] - [78], but at a much greater cost in terms of computation. These two schemes are complementary and are commonly used jointly in contemporary rendering systems. Specialist computing hardware has also led to significant progress in graphics-based generative models. In 1999, the first Graphics Processing Unit (GPU) was introduced and transformed the way rendering worked as it allowed rasterization to be done in parallel. The viability of using artificial neural networks on GPUs was later demonstrated by Oh Kyoung-Su and Jung Keechul [81], which was an important milestone to modern GPUs-accelerated deep learning. The GAI systems have become indispensable in terms of hardware components being used over the years. Also, standard graphics programming interfaces like OpenGL [79] and Direct3D [80] have facilitated hardware- Independent development and simplified rendering pipelines.

Generative methods using graphics have been widely used in computer animation and visual media. In 1958, the earliest computer-generated movie was released and ushered in the use of generative graphics in media creation. The breakthrough of the first fully computer- generated feature film Toy Story was a milestone in showing the commercial viability of the graphics-based generation. These techniques have since become the key to industries like film making, video games and virtual reality allowing highly realistic environments and advanced visual effects to be produced.

DEEP GENERATIVE METHODOLOGIES

The renaissance of Deep Learning [82],[83] has played a major role in significant progress in generative artificial intelligence (GAI). Increasing the depth of neural networks around 2011 pioneering studies [2] confirmed that the deeper the neural network, the better it was able to learn hierarchical data representations, performing almost or even better than humans in classification tasks. This was also confirmed by later studies [3], [84], which made deep neural networks a leader in artificial intelligence. Deep neural networks have since found widespread application in generative problems [4], [85], and have shown impressive abilities in the modeling of complicated data distributions and the production of very realistic results.

A. Deep Network Architectures

The work on deep neural network architectures to model generative processes can be broadly divided into two directions: the improvement of recurrent neural network (RNN)-based models and the developments that are fueled by attention mechanisms.

Innovations in better RNN structures enabled the first advancements in sequence modeling. As an example, Felix Gers et al. [86] proposed a type of recurrent networks that have forget gates to more effectively handle long-term dependencies. This idea was developed into the concept of gated architectures like Gated Recurrent Unit (GRU) [87], which use update and reset gates to control the flow of information. The extent to which the past information is retained is determined by the update gate and the discarding of past information is determined by the reset gate. These architectures are computationally efficient and suited to real-time generative applications, but have limitations in the ability to capture very long-range dependencies.

One of the significant advancements in deep learning architecture is the introduction of the Transformer architecture [88]. The input of data in transformers is in the form of tokens which are the basic units of data. In natural language processing, tokens can be words, characters, or sub word units, and again, this is determined by the tokenization strategy [89]91]. Models that use transformers enable working with multimodal data, such as text, images, and audio, which makes them state-of-the-art in addressing a variety of generative tasks.

Transformers are based on two processes positional encoding and self-attention. Positional encoding provides sequence-order information to input embeddings, and allows the model to learn the structure of sequential data. The self-attention mechanism attaches the adaptive weights to various inputs enabling the model to put emphasis on the most important information. This design allows transformers to successfully learn long- range dependencies which is a major strength compared to the traditional RNN- based architectures. Moreover, transformers are very well-suited to parallel computation in GPUs leading to effective training and inference.

Recent studies have generalized transformer structures to computer vision work. Vision transformers (ViTs) [90], [92] split images into fixed-size patches and encode them into a sequence of tokens, thus harmonizing the processing architecture of visual and textual data. Moreover, a few papers have worked on how to enhance the efficiency of transformers, such as linearized attention models [93], [94], sparse attention models [95], and approximation models [96].

In addition to transformers, other architecture types have been inspired by attention mechanisms. As an example, graph attention networks [97] are useful at modelling structured data like social networks and protein interaction graphs. Capsule

Network [98] and State Space Model [99] are other emergent architectures that are intended to maintain hierarchical relationships in data and scalable frameworks of long-sequence modelling, respectively.

Deep Generative Models

Deep generative models are a type of machine learning models that are based on deep neural networks to learn complex data distributions and produce new samples. These models are based on different generative theories, and the most popular ones are Generative Adversarial Network (GAN), Variational Autoencoder (VAE), and Diffusion Model.

A. Generative Adversarial Networks (GANs)

This has been the case with generative adversarial networks (GANs) [100], [101] as they are able to produce extremely realistic data. GANs are comprised of two rival neural networks; a discriminator and a generator. The generator is supposed to make synthetic samples, which are similar to real data whereas the discriminator tries to differentiate between the real and the generated samples. This is modeled as a minimax game, with both networks playing improved to a Nash equilibrium, the generated samples are then indistinguishable to real data.

GAN-based models [102]–[105] have some important benefits regarding controllability and high-quality synthesis. As an example, StyleGAN [104] allows controlling images at the pixel level with semantics that are in fine-grained mode. Moreover, GANs are computationally inexpensive to infer as opposed to conventional graphics-based algorithms. Nonetheless, GANs are affected by a famous drawback mode collapse, the inability of the generator to produce all the diversity of the data distribution. Training instability is a significant problem, in spite of a number of solutions [106] that have been suggested to alleviate this problem, large-scale models are particularly a challenge.

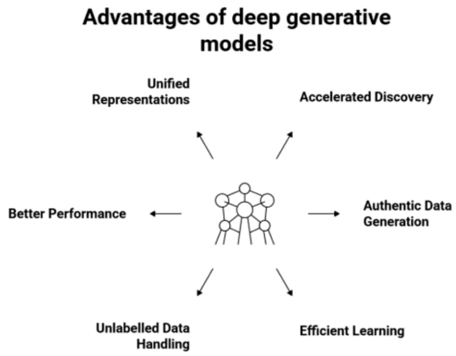

Figure_2: Deep Generative Methodologies

B. Variational Autoencoders (VAEs)

Another notable deep generative model that learns data distributions in a latent space is the variational autoencoder (VAE) [107]. A VAE is made up of two primary parts: an encoder which converts the input data into a latent representation, and a decoder which re-creates the data in terms of that latent space. The model, during training, minimizes reconstruction error but constrained by the latent distribution to approximate a predefined prior distribution.

VAE-based models [108], [109] are a complement to GANs that are more exhaustive of the data distribution. The latent space can be sampled to generate various and hitherto unknown data points. But VAEs generally yield less sharp and blurrier outputs than GAN-generated ones, as the results are obtained through a probabilistic reconstruction process.

C. Probabilistic Diffusion Models

The most recent advancement in generative modeling is probabilistic diffusion models [37], [38]. These models consider data generation as a stochastic process with two steps: forward diffusion process and reverse denoising process. Noise in the forward process is progressively injected into real data until it becomes indistinguishable to random noise. This is then reversed using the model that is trained to remove noise to rebuild meaningful data.

The most recent studies have been working towards enhancing the performance of diffusion models by techniques like latent diffusion [110], introduction of discriminative priors [111], and model distillation [112]. Diffusion models have shown remarkable results in various generative tasks, especially in zero-shot environments, where they are capable of creating new and very realistic images. Nevertheless, such models demand large computational power to be trained and inferred, and in many cases, they surpass the needs of GAN-based ones.

D. Applications and Impact

Deep generative models, being more efficient in high-dimensional data, have been substituting traditional methods of statistics since the late 2010s. They have been able to produce unprecedented flexibility and realism due to their capability to utilize large-scale datasets. In most instances, the content that is created by GAI has become unrecognizable to real world information.

Transformer based generative models have also broadened the area of application, such as language translation, document generation and code synthesis. Audio models, like Wave Net [113], have been shown to be able to produce highly realistic speech and music. Also, programs like Face Swap [114] demonstrate the possibilities of deep generative models in manipulating media, both generating and posing challenges and raising ethical issues.

Deep Generative Grammar of Graphics

The effective use of deep neural networks in three-dimensional (3D) perception and scene understanding has motivated the desire to combine learning-based models with conventional rendering methods. But classical rendering pipelines are generally not differentiable in terms of model parameters, so they cannot be optimized using gradient-based methods. To overcome this limitation, recent studies have been devoted to differentiable rendering frameworks that allow end-to- end learning and optimizing representations of 3D scenes directly. Differentiable rendering makes the gradient flow forward through the rendering process, and neural networks to learn scene geometry, appearance, and lighting in one network. The most notable one is the Neural Radiance Field (NeRF) [7], a framework of implicit neural representations of scenes. This method is based on the concepts of ray tracing, in which the multi-layer perceptron is trained to approximate a continuous volumetric field that characterizes the scene. Using volume rendering methods, NeRF approximates how light interacts with the scene, allowing the generation of photorealistic images of arbitrary viewpoints.

Later research has tried to make neural rendering computationally efficient. An example is the sparse voxel representations proposed by Guy Fridovich-Keil et al. [115] to render images fast but with high visual quality. More recently, an alternative has appeared, 3D Gaussian Splatting [116], which uses rasterization pipelines and neural point-based representations, and is also efficient. This approach allows real-time execution whilst maintaining high-quality image synthesis, and is especially applicable in interactive applications. Improved hardware has helped in the use of deep generative learning in graphics as well. Realistic rendering has been made possible by recent advances in ray tracing algorithms, made possible by the modern GPUs. These techniques enabled the production of very realistic digital avatars in commercial movies during the 2010s. In addition, the visual integration has been facilitated by the hybrid rendering methods where computer-generated imagery is integrated with a live-action footage. Super-sampling using deep learning methods have also increased the efficiency of rendering as they allow real-time ray tracing at high resolutions, including 4K, on consumer- level GPUs. The developments have been instrumental in the creation of immersive technologies, such as virtual reality (VR), augmented reality (AR), and mixed reality (MR). Consequently, the deep generative learning has become a pillar of the creation of interactive and photorealistic digital environments.

FUTURE DIRECTIONS (with IEEE Citations)

Generative artificial intelligence (GAI) applications built upon foundation models have become the dominant paradigm in recent years. However, these models inherit several vulnerabilities, and research on GAI safety continues to lag behind rapid technological advancements. In this section, we outline critical challenges and promising future directions for the development of safe, reliable, and scalable GAI systems.

A. Safety and Governance Challenges Value alignment:

A fundamental challenge in GAI is ensuring alignment with human values and intentions. Models must generate content that is helpful, ethical, and resistant to misuse. Achieving this objective requires robust evaluation frameworks capable of measuring alignment performance [161]. Current evaluation methods, largely based on statistical metrics, are insufficient to capture nuanced human preferences. Therefore, future research must focus on developing comprehensive alignment strategies and guidelines that reflect diverse societal values.

Source identification:

The increasing realism and manipulability of GAI-generated content raise concerns regarding authenticity and intellectual property. It is essential to develop reliable mechanisms for tracing the origin of generated content. Techniques such as digital watermarking and cryptographic signatures are promising solutions; however, embedding robust and tamper-resistant identifiers without degrading usability remains a significant technical challenge.

Security regulations:

The rapid deployment of GAI technologies necessitates the establishment of regulatory frameworks to ensure ethical development and usage. Organizations such as IEEE and ISO play a crucial role in defining standards for safe AI practices. Developers must adhere to these guidelines, ensuring that their systems do not cause harm. Additionally, mechanisms for auditing, correcting, or retracting harmful models should be implemented, analogous to the peer-review and retraction processes in scientific research.

B. Emerging Research Directions Unification of modalities:

Although recent advancements have enabled integration between text and image modalities, the fusion of multiple modalities—including text, images, video, audio, and structured data—remains an open challenge. Future research must address cross-modal alignment and interaction among heterogeneous systems [162], enabling seamless multimodal reasoning and generation.

Deciphering GAI models:

Understanding the internal mechanisms of GAI systems is essential for transparency and trust. Current interpretability approaches [163] provide heuristic insights but lack rigorous theoretical grounding. Integrating principles from physics-inspired domains such as thermodynamics [38] and Electrodynamics [164] offers promising directions for improving model interpretability. However, these approaches must be formalized within computational frameworks to ensure reliability and reproducibility.

Learning from synthetic data:

The growing availability of synthetic data generated by GAI systems presents new opportunities for machine learning. Such data can reduce labeling costs, augment existing datasets, and support knowledge transfer. While traditionally viewed as interpolations of existing patterns, modern GAI systems can generate novel and meaningful data representations. Consequently, synthetic data may become a dominant resource for training future AI systems.

Supervision beyond human capability:

Historically, generative models have relied on human-curated data and supervision. However, recent advancements in reinforcement learning have demonstrated that models can achieve high-level reasoning capabilities without explicit human supervision [165]. This raises the possibility of developing systems with superhuman performance through self-improvement. As these capabilities evolve, it becomes critical to design new learning paradigms that ensure such systems remain aligned with human interests and are subject to effective regulatory control.

CONCLUSION

This work has presented a comprehensive overview of the historical evolution and current advancements in generative artificial intelligence (GAI). The methodologies were systematically categorized into four major paradigms: rule-based generative systems, model- based generative algorithms, deep generative methodologies, and foundation models. Each paradigm was analyzed in terms of its underlying principles, technological characteristics, and practical applications. Rather than exhaustively reviewing all existing literature, this study emphasized representative approaches that illustrate the fundamental ideas and developmental trends in GAI. The transition from symbolic rule-based systems to data-driven and learning-based approaches highlights the progressive shift toward scalability, adaptability, and generalization. In particular, the emergence of Deep Learning and Transformer architecture has enabled significant breakthroughs in generative capabilities, culminating in the development of large-scale foundation models such as ChatGPT. These models demonstrate remarkable performance across diverse domains, including text, image, audio, and multimodal content generation. Despite these advancements, several challenges remain. Issues related to model interpretability, ethical alignment, computational efficiency, and security continue to pose significant barriers to the safe and sustainable deployment of GAI technologies. This work has highlighted these concerns and discussed emerging research directions aimed at addressing them, including multimodal integration, explainability, and the use of synthetic data.In conclusion, GAI represents a rapidly evolving field with transformative potential across numerous disciplines. The methodologies and strategies discussed in this paper provide a foundation for future research and innovation. As the field progresses, continued efforts in addressing its limitations and risks will be essential to ensure that GAI systems are developed and deployed in a manner that benefits society as a whole.

Reference:

1. OpenAI. (2023). *ChatGPT: Optimizing language models for dialogue*. https://openai.com
2. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444. https://doi.org/10.1038/nature14539
3. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
4. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language models are unsupervised multitask learners* (Technical report). OpenAI.
5. Van den Oord, A., *et al.* (2016). *WaveNet: A generative model for raw audio*. In *Proceedings of the Speech Synthesis Workshop (SSW)*.
6. Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

7. Zhang, R., *et al.* (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
8. Vaswani, A., *et al.* (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*.
9. Radford, A., *et al.* (2018). *Improving language understanding by generative pre-training*. OpenAI.
10. OpenAI. (2023). *GPT-4 technical report*.
11. Tesla, Inc. (2022). *Autopilot and full self-driving capabilities*.
12. DeepMind. (2021). AlphaFold: Using AI for protein structure prediction. *Nature*.
13. Feigenbaum, E. A. (1980). Artificial intelligence research: The emerging discipline. *AI Magazine, 1*(1), 13–24.
14. Newell, A., & Simon, H. A. (1972). *Human problem solving*. Prentice Hall.
15. Weizenbaum, J. (1966). ELIZA—A computer program for the study of natural language communication. *Communications of the ACM, 9*(1), 36–45.
16. McCarthy, J. (1959). Programs with common sense. In *Proceedings of the Teddington Conference*.
17. Shortliffe, E. H. (1976). *Computer-based medical consultations: MYCIN*. Elsevier.
18. Colby, K. M. (1971). Artificial paranoia: A computer simulation of paranoid processes. *Artificial Intelligence, 2*(1), 1–25.
19. SYSTRAN. (1968–2007). *SYSTRAN translation system*.
20. Fant, G. (1960). *Acoustic theory of speech production*. Mouton.
21. Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
22. Koller, D., & Friedman, N. (2009). *Probabilistic graphical models*. MIT Press.
23. Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm.
24. Markov, A. A. (1906). Theory of Markov processes.
25. Baum, L. E., *et al.* (1960s). Hidden Markov models.
26. Harris, T. E. (1978). Additive set-valued Markov processes and graphical methods. *The Annals of Probability, 6*(3), 355–378.
27. Darroch, J. N., Lauritzen, S. L., & Speed, T. P. (1980). Markov fields and log-linear interaction models for contingency tables. *The Annals of Statistics, 8*(3), 522–539.
28. Li, S. Z. (1994). Markov random field models in computer vision. In J. O. Eklundh (Ed.), *Computer Vision—ECCV '94* (pp. 361–370). Springer.
29. Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology, 7*(3–4), 601–620.
30. Jurafsky, D., & Martin, J. H. (2000). *Speech and language processing*. Prentice Hall.
31. Schmidhuber, J., & Heil, S. (1996). Sequential neural text compression. *IEEE Transactions on Neural Networks, 7*(1), 142–146.
32. Abresch, U., & Langer, J. (1986). The normalized curve shortening flow and homothetic solutions. *Journal of Differential Geometry, 23*(2), 175–196.
33. Stein, C. (1972). A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 2, pp. 583–602).
34. Song, Y., Sohl-Dickstein, J., Kingma, D. P., *et al.* (2023). Score-based generative modeling through stochastic differential equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
35. Zhang, Q., & Chen, Y. (2021). Diffusion normalizing flow. In *Proceedings of the 35th International Conference on Neural Information Processing Systems (NeurIPS)* (pp. 16280–16291). Curran Associates.
36. Van Kampen, N. G. (1976). Stochastic differential equations. *Physics Reports, 24*(3), 171–228.
37. Song, Y., & Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS)* (pp. 11918–11930). Curran Associates.
38. Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)* (pp. 6840–6851). Curran Associates.
39. Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., *et al.* (2023). Flow matching for generative modeling. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
40. McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics, 5*(4), 115–133.
41. Stigler, S. M. (1981). Gauss and the invention of least squares. *The Annals of Statistics, 9*(3), 465–474.
42. Merriman, M. (1877). *A list of writings relating to the method of least squares: With historical and critical notes*. Academy Press.
43. Jenkin, N. (1957). Affective processes in perception. *Psychological Bulletin, 54*(2), 100–127.
44. Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review, 65*(6), 386–408.

45. Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics, 36*(4), 193–202.
46. Zhang, W., Tanida, J., Itoh, K., & Ichioka, Y. (1988). Shift-invariant pattern recognition neural network and its optical architecture. In *Proceedings of the Annual Conference of the Japan Society of Applied Physics* (pp. 2147–2151).
47. Lenz, W. (1920). Beitrag zum Verständnis der magnetischen Erscheinungen in festen Körpern. *Zeitschrift für Physik, 21*, 613–615.
48. Ising, E. (1924). *Beitrag zur Theorie des Ferro- und Paramagnetismus* (Doctoral dissertation, University of Hamburg, Germany).
49. Kleene, S. C. (1951). Representation of events in nerve nets and finite automata. In *Automata Studies* (Vol. 1, pp. 3–103).
50. Amari, S.-I. (1972). Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Transactions on Computers, C-21*(11), 1197–1206.
51. Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, 79*(8), 2554–2558.
52. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.
53. Heilmann, P., & Rigney, D. (1981). An energy-based model of friction and its application to coated systems. *Wear, 72*(2), 195–217.
54. Bakır, G. H., Hofmann, T., Schölkopf, B., Smola, A. J., Taskar, B., & Vishwanathan, S. V. N. (2007). *Predicting structured data*. MIT Press.
55. Rumelhart, D. E., & McClelland, J. L. (1986). *Information processing in dynamical systems: Foundations of harmony theory*. MIT Press.
56. Linnainmaa, S. (1970). *The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors* (Master's thesis, University of Helsinki, Finland).
57. Werbos, P. J. (1982). Applications of advances in nonlinear sensitivity analysis. In R. F. Drenick & F. Kozin (Eds.), *System modeling and optimization* (pp. 762–770). Springer.
58. 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
59. LeCun, Y., Touresky, D., Hinton, G., *et al.* (1988). A theoretical framework for back-propagation. In *Proceedings of the Connectionist Models Summer School* (pp. 21–28). Morgan Kaufmann.
60. Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.
61. Peskun, P. H. (1973). Optimum Monte Carlo sampling using Markov chains. *Biometrika, 60*(3), 607–612.
62. Geyer, C. J. (1992). Practical Markov chain Monte Carlo. *Statistical Science, 7*(4), 473–483.
63. Brooks, S. (1998). Markov chain Monte Carlo method and its application. *Journal of the Royal Statistical Society: Series D (The Statistician), 47*(1), 69–100.
64. Chib, S., & Greenberg, E. (1995). Understanding the Metropolis–Hastings algorithm. *The American Statistician, 49*(4), 327–335.
65. Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research, 15*(1), 1593–1623.
66. Chen, T., Fox, E., & Guestrin, C. (2014). Stochastic gradient Hamiltonian Monte Carlo. In *Proceedings of the 31st International Conference on Machine Learning (ICML)* (pp. 1683–1691). PMLR.
67. Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning, 37*(2), 183–233.
68. Wainwright, M. J., & Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning, 1*(1–2), 1–305.
69. Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. MIT Press.
70. Mildenhall, B., *et al.* (2020). NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*.
71. Point cloud. (n.d.). Standard 3D representation technique.
72. Voxel. (n.d.). Volumetric representation method.
73. Catmull, E. (1974). A subdivision algorithm for computer display.
74. Blinn, J. (1978). Simulation of wrinkled surfaces.
75. Khronos Group. (1992). *OpenGL specification*.
76. Appel, A. (1968). Some techniques for shading machine renderings.
77. T. Whitted, "An improved illumination model for shaded display," 1980.
78. NVIDIA, "Real-time ray tracing technologies," 2018.
79. Khronos Group, *OpenGL*.
80. Microsoft, *Direct3D*.
81. K.-S. Oh and K. Jung, "GPU implementation of neural networks," 2004.
82. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, 2015.

83. Geoffrey Hinton et al., “Reducing the dimensionality of data with neural networks,” *Science*, 2006.
84. Alex Krizhevsky et al., “ImageNet classification with deep convolutional neural networks,” *NeurIPS*, 2012.
85. Diederik P. Kingma and Max Welling, “Auto-encoding variational Bayes,” 2014.
86. F. Gers et al., “Learning to forget: Continual prediction with LSTM,” 2000.
87. K. Cho et al., “Learning phrase representations using RNN encoder–decoder,” 2014.
88. Ashish Vaswani et al., “Attention is all you need,” *NeurIPS*, 2017.
89. Tokenization, standard NLP technique.
90. A. Dosovitskiy et al., “An image is worth 16×16 words: Transformers for image recognition,” 2020.
91. Byte Pair Encoding, subword tokenization method.
92. Z. Dai et al., “Transformer-XL: Attentive language models,” 2019.
93. S. Wang et al., “Linformer: Self-attention with linear complexity,” 2020.
94. K. Choromanski et al., “Performer: Linear attention,” 2021.
95. R. Child et al., “Sparse transformers,” 2019.
96. I. Beltagy et al., “Longformer: Efficient transformers,” 2020.
97. P. Veličković et al., “Graph attention networks,” 2018.
98. S. Sabour et al., “Dynamic routing between capsules,” 2017.
99. A. Gu et al., “State space models for sequence modeling,” 2021.
100. Ian Goodfellow et al., “Generative adversarial nets,” *NeurIPS*, 2014.
101. M. Mirza and S. Osindero, “Conditional generative adversarial nets,” 2014.
102. T. Karras et al., “Progressive growing of GANs,” 2018.
103. H. Zhang et al., “StackGAN,” 2017.
104. T. Karras et al., “StyleGAN: A style-based generator architecture,” 2019.
105. A. Brock et al., “Large scale GAN training,” 2019.
106. T. Salimans et al., “Improved techniques for training GANs,” 2016.
107. Kingma, D. P., & Welling, M. (2014). *Auto-encoding variational Bayes*. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
108. D. J. Rezende et al., “Stochastic backpropagation and approximate inference in deep generative models,” 2014.
109. I. Higgins et al., “β-VAE: Learning basic visual concepts with a constrained variational framework,” 2017.
110. Robin Rombach et al., “High-resolution image synthesis with latent diffusion models,” *CVPR*, 2022.
111. Prafulla Dhariwal and Alexander Nichol, “Diffusion models beat GANs on image synthesis,” *NeurIPS*, 2021.
112. J. Ho et al., “Denoising diffusion probabilistic models,” *NeurIPS*, 2020.
113. Aäron van den Oord et al., “WaveNet: A generative model for raw audio,” 2016.
114. FaceSwap, open-source project, 2017.
115. G. Fridovich-Keil et al., “PlenOctrees for real-time rendering of neural radiance fields,” 2021.
116. B. Kerbl et al., “3D Gaussian splatting for real-time radiance field rendering,” 2023.
117. T. Karras, M. Aittala, T. Aila, and S. Laine, “Alias-free generative adversarial networks,” *NeurIPS*, vol. 34, pp. 852–863, 2021.
118. A. Nichol and P. Dhariwal, “Improved denoising diffusion probabilistic models,” in *ICML*, 2021, pp. 8162–8171.
119. R. Rombach et al., “High-resolution image synthesis with latent diffusion models,” in *CVPR*, 2022, pp. 10684–10695.
120. C. Saharia et al., “Photorealistic text-to-image diffusion models with deep language understanding,” *NeurIPS*, vol. 35, pp. 36479–36494, 2022.
121. A. Radford et al., “Learning transferable visual models from natural language supervision,” in *ICML*, 2021, pp. 8748–8763.
122. T. Brown et al., “Language models are few-shot learners,” *NeurIPS*, vol. 33, pp. 1877–1901, 2020.
123. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL-HLT*, 2019, pp. 4171–4186.
124. A. Vaswani et al., “Attention is all you need,” *NeurIPS*, vol. 30, pp. 5998–6008, 2017.
125. Ian Goodfellow, Yoshua Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
126. Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE TPAMI*, vol. 35, no. 8, pp. 1798–1828, 2013.
127. D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” in *ICLR*, 2014.
128. I. Goodfellow et al., “Generative adversarial nets,” *NeurIPS*, vol. 27, pp. 2672–2680, 2014.
129. S. Reed et al., “Generative adversarial text-to-image synthesis,” in *ICML*, 2016, pp. 1060–1069.
130. H. Zhang, T. Xu, H. Li, et al., “StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks,” in *ICCV*, 2017, pp. 5907–5915.
131. T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *CVPR*, 2019, pp. 4401–4410.
132. J. Ho and T. Salimans, “Classifier-free diffusion guidance,” *NeurIPS*, 2021.
133. P. Dhariwal and A. Nichol, “Diffusion models beat GANs on image synthesis,” *NeurIPS*, vol. 34, pp. 8780–8794, 2021.

134. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.
135. C. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *ICLR*, 2021.
136. A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," *NeurIPS*, 2016.
137. M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv:1411.1784, 2014.
138. X. Mao et al., "Least squares generative adversarial networks," in *ICCV*, 2017, pp. 2794–2802.
139. T. Salimans et al., "Improved techniques for training GANs," *NeurIPS*, 2016.
140. M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," in *ICML*, 2017, pp. 214–223.
141. I. Gulrajani et al., "Improved training of Wasserstein GANs," *NeurIPS*, 2017.
142. Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.
143. Sepp Hochreiter et al., "Gradient flow in recurrent nets: The difficulty of learning long-term dependencies," 2001.
144. J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *EMNLP*, 2014, pp. 1532–1543.
145. Tomas Mikolov et al., "Efficient estimation of word representations in vector space," arXiv:1301.3781, 2013.
146. O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015, pp. 234–241.
147. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.
148. C. Szegedy et al., "Going deeper with convolutions," in *CVPR*, 2015, pp. 1–9.
149. Geoffrey Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv:1503.02531, 2015.
150. Richard Sutton and Andrew Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
151. D. Silver et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, 2016.
152. D. Silver et al., "Mastering the game of Go without human knowledge," *Nature*, vol. 550, pp. 354–359, 2017.
153. K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *ICCV*, 2017, pp. 2961–2969.
154. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016, pp. 779–788.
155. A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *NeurIPS*, 2012.
156. R. Girshick, "Fast R-CNN," in *ICCV*, 2015, pp. 1440–1448.
157. J. Howard and S. Gugger, "Fastai: A layered API for deep learning," *Information*, 2020.
158. OpenAI, "GPT-4 technical report," 2023.
159. Google DeepMind, "Gemini: A family of highly capable multimodal models," 2023.
160. Anthropic, "Claude: Constitutional AI and alignment techniques," 2023.
161. OpenAI, "Aligning AI systems with human values," 2023.
162. A. Radford et al., "Learning transferable visual models from natural language supervision," in *ICML*, 2021.
163. B. Kim et al., "Interpretability beyond feature attribution," 2018.
164. James Clerk Maxwell, *A Treatise on Electricity and Magnetism*. Oxford, U.K.: Clarendon Press, 1873.
165. DeepMind, "Advances in reinforcement learning for reasoning," 2024.
166. Yann LeCun, "A path towards autonomous machine intelligence," 2022.
167. Stuart Russell, *Human Compatible: Artificial Intelligence and the Problem of Control*. New York, NY, USA: Viking, 2019.
168. Nick Bostrom, *Superintelligence: Paths, Dangers, Strategies*. Oxford, U.K.: Oxford University Press, 2014.