IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

A Comprehensive Survey of Generative AI: Applications and Future Directions Across Domains

AMIR JAVADPOUR[1], FOROUGH JA'FARI[2], TARIK TALEB[3] (Member, IEEE), and Chafika Benzaïd [4]
[1]Senior Cybersecurity Researcher MOSA!C Lab / ICTFICIAL Oy, Finland.
[2]Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
[3]Faculty of Electrical Engineering and Information Technology, Ruhr University Bochum, Bochum
[4]Faculty of Information Technology and Electrical Engineering, University of Oulu

Corresponding author: Amir Javadpour (e-mail: a.javadpour87@gmail.com).

ABSTRACT Generative Artificial Intelligence (GenAI) has become a foundational paradigm for learning complex data distributions and synthesizing realistic content across text, image, audio, tabular, scientific, and multimodal data. Despite rapid progress, the literature remains fragmented across model families and application domains, making it difficult to compare methodological choices, evaluation practices, and domain-specific deployment constraints. This survey addresses this gap by reviewing GenAI from both an algorithmic and cross-domain perspective. Following a PRISMA-inspired search and screening procedure, we organize representative and technically relevant studies across major GenAI families, including variational autoencoders, generative adversarial networks, diffusion models, autoregressive models, flow-based models, and recent foundation and multimodal models. We synthesize applications across healthcare, agriculture, manufacturing, transportation, earth and environmental systems, computer science and networks, materials science, finance and economics, education, business and services, and creative industries. The review shows that diffusion and autoregressive foundation models increasingly dominate high-fidelity image, language, and multimodal generation, while GANs, VAEs, and flow-based models remain important in data-limited, structured, scientific, and privacy-aware settings. Beyond summarizing applications, the paper compares domain-specific data structures, evaluation practices, robustness concerns, failure modes, and ethical risks. The survey contributes a unified reviewed abstraction of GenAI workflows, a comparison with prior surveys, a domain-aware synthesis of open gaps, and a future roadmap for trustworthy, validated, and responsible GenAI deployment.

INDEX TERMS Generative Artificial Intelligence (GenAI), Variational Autoencoder (VAE), Generative Adversarial Network (GAN), Diffusion Probabilistic Model, Autoregressive Model, Flow-Based Generative Model.

I. INTRODUCTION

IN an era defined by unprecedented volumes of digital information and rapidly evolving socio-technical systems, the ability to generate, interpret, and reshape data has become a cornerstone of innovation. Today, societies are increasingly shaped by intelligent infrastructures, ranging from global communication networks and interconnected industries to data-driven healthcare ecosystems and autonomous transportation platforms. As these systems continue to expand in scale and complexity, traditional analytical and predictive tools are no longer sufficient to cope with the richness, diversity, and dynamism of modern data environments [1].

Against this backdrop, Generative Artificial Intelligence (GenAI) has emerged as one of the most transformative technological developments of the past decade. Unlike conventional machine learning methods that primarily classify or predict outcomes, GenAI models possess the remarkable capacity to create: to synthesize new information, simulate intricate patterns, adapt to previously unseen conditions, and construct high-dimensional representations that mirror, extend, or augment real-world phenomena. These generative capabilities have catalyzed new research paradigms by enabling machines not merely to understand data, but to imagine within data .

This shift has had profound implications across the scientific, industrial, and societal domains. Healthcare systems leverage generative models to simulate patient data and support diagnostic decision-making; agricultural platforms employ them to optimize crop management; manufacturing lines use them to design components and improve automation; environmental scientists rely on them to model climate behavior; financial institutions adopt them to assess risks and identify market patterns; and digital networks integrate them to manage traffic, detect anomalies, and enhance security. The breadth of these applications signals not only the versatility of GenAI but also its potential to redefine foundational practices across disciplines [2, 3].

Despite this growing influence, the landscape of GenAI research remains fragmented across domains, each adopting generative models to address unique challenges. While numerous studies explored specific algorithms or isolated applications, such as the work done by Rabbani et al. [4] focusing on healthcare, a comprehensive, cross-domain understanding of how GenAI is currently deployed and what opportunities lie ahead has yet to be fully synthesized. Such consolidation is essential for illuminating broader trends, identifying methodological gaps, and offering future research directions that meaningfully link technological progress with real-world needs.

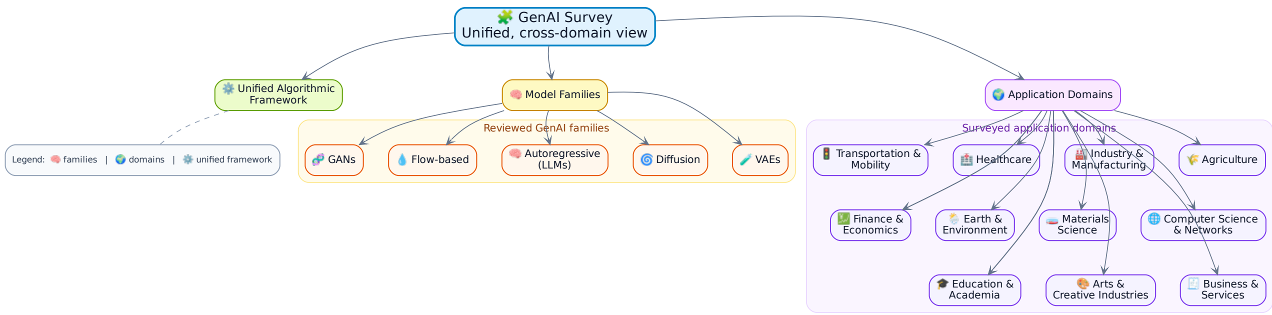

FIGURE 1. **Overview of this survey: a unified GenAI framework, major model families, and the cross-domain application areas reviewed.**

Motivated by this necessity, the present survey provides a systematic and domain-oriented examination of GenAI applications across major sectors, including healthcare, agriculture, industry and manufacturing, transportation, environment, computer science and networks, material science, finance and economics, education, and business. The survey not only synthesizes existing research, but also delineates emerging opportunities and underexplored areas where generative models hold significant promise.

Existing surveys on GenAI, such as those by Sengar et al. [5], Gozalo-Brizuela and Garrido-Merchán [6], and Zhang et al. [7], primarily provide high-level taxonomies of GenAI models and broad overviews of application domains, but do not offer a systematic, domain-specialized analysis that connects algorithmic mechanisms with concrete implementation workflows. Moreover, these works generally describe applications at a conceptual level and lack detailed methodological bridges that practitioners need to reproduce or adapt generative models in real settings. Finally, prior surveys tend to treat application domains homogeneously, overlooking cross-domain differences in data structure, constraints, and evaluation metrics, which limits their usefulness for researchers seeking tailored guidance within specific scientific or industrial contexts.

A. SURVEY METHODOLOGY AND SCOPE

This survey follows a PRISMA-inspired review strategy to improve transparency and reproducibility while retaining the flexibility required for a broad, cross-domain GenAI survey. The literature search was conducted across major scientific databases and digital libraries, including IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, Web of Science, Scopus, PubMed, and arXiv. The search combined model-family terms such as "generative artificial intelligence," "generative model," "variational autoencoder," "generative adversarial network," "diffusion model," "autoregressive model," "large language model," "foundation model," "multimodal generative model," and "normalizing flow" with domain terms including healthcare, agriculture, manufacturing, transportation, environment, computer networks, cybersecurity, materials science, finance, education, business, and creative industries.

The inclusion criteria were: 1) studies that use or review a generative model as a central methodological component; 2) works that provide a clear application, evaluation, or technical discussion relevant to at least one surveyed domain; 3) peer-reviewed journal, conference, or book-chapter publications, with selected high-impact preprints included only when they represent widely used foundation models, benchmark systems, or emerging topics not yet covered by peer-reviewed literature; and 4) publications written in English. The exclusion criteria were: 1) works where generative models are only mentioned superficially; 2) papers without sufficient methodological or application detail; 3) duplicate or substantially overlapping versions of the same work; and 4) non-technical opinion pieces, undocumented web resources, and sources without identifiable scholarly provenance.

The screening process involved title/abstract inspection, full-text relevance checking, and thematic coding by model family, data type, application task, evaluation strategy, limitation, and future research gap. Because the paper aims to synthesize a rapidly expanding cross-domain field rather than conduct a narrow meta-analysis, quantitative PRISMA counting is used as a guiding principle rather than as the sole organizing mechanism. The final organization emphasizes representative and technically relevant studies that clarify methodological trends, domain constraints, and cross-domain patterns.

Table 1 summarizes the main differences between representative prior GenAI surveys and the present survey, highlighting the added methodological transparency, domain-aware synthesis, and responsible-deployment analysis provided in this work.

Through this structured exploration, the paper seeks to offer both a panoramic view of the current GenAI landscape and a forward-looking perspective that supports researchers, practitioners, and policymakers in navigating the evolving frontier of generative technologies. The key contributions of this paper are:

- **We provide a unified reviewed abstraction of GenAI workflows**, capturing the common computational stages, modeling assumptions, and training/generation procedures shared across major generative families. The abstraction is not intended as a new theoretical algorithm, but as a survey-oriented organizing lens that helps compare VAEs, GANs, diffusion models, autoregressive architectures, and flow-based models under consistent notation and workflow stages.
- **We present a broad multi-domain synthesis of GenAI applications**, covering scientific, industrial, and societal sectors. For each domain, we review representative studies and connect them to data structures, task settings, model families, evaluation practices, and deployment constraints. This enables readers to understand not only where GenAI is used, but also why particular model families are suitable or limited in each domain.
- **We critically discuss limitations, risks, and future directions**, including methodological failure modes, evaluation weaknesses, privacy and fairness concerns, hallucination risks, and domain-specific validation challenges. The resulting roadmap identifies near-term, mid-term, and long-term research priorities for trustworthy and responsible GenAI across established and emerging application areas.

Figure 1 provides a compact roadmap of this survey. We organize the review around a unified algorithmic framework for generative modeling, then relate it to the main GenAI families (GANs, VAEs, diffusion, autoregressive, and flow-based models), and finally synthesize the literature across the major application domains covered in this work. This map is intended to help readers quickly locate where a given domain fits in the overall narrative and how domain-specific practices connect back to common generative building blocks.

Fig. 2 further expands the introductory roadmap by presenting the survey as a layered structure. The first layer summarizes the core

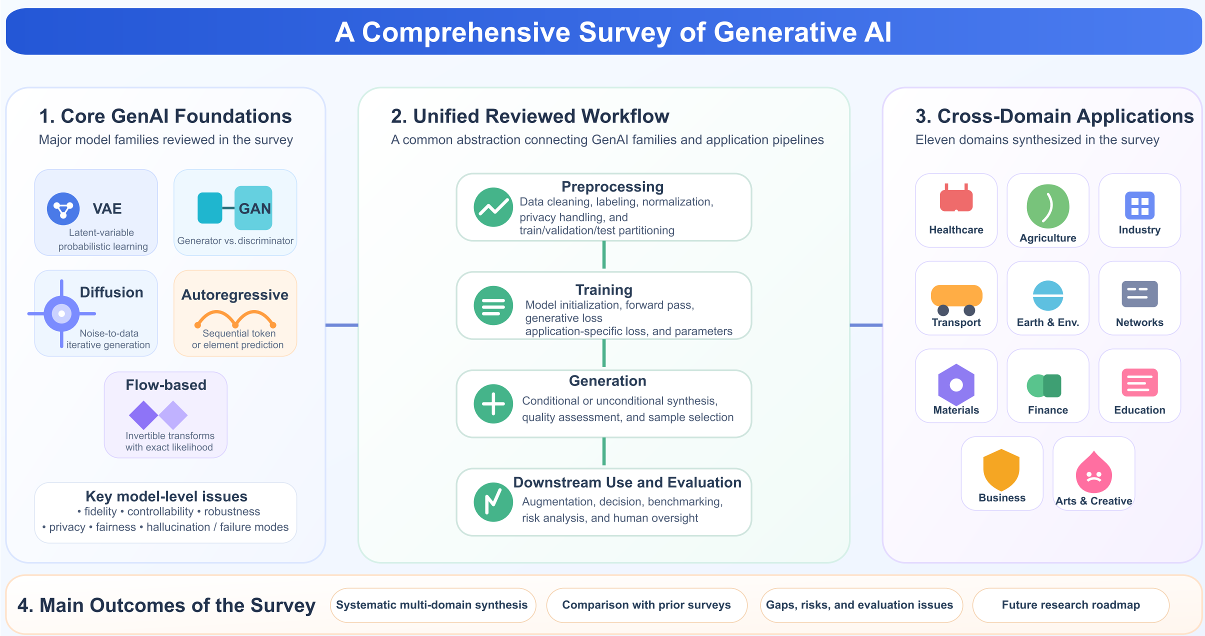

FIGURE 2. Layered overview of the survey structure, linking GenAI model families, the reviewed workflow, application domains, and key survey outputs.

GenAI model families, including VAEs, GANs, diffusion models, autoregressive models, and flow-based models. The second layer highlights the reviewed workflow adopted throughout the paper, covering data inputs, preprocessing, training, generation, evaluation, and downstream usage. The third layer maps these methodological foundations to the major application domains reviewed in this survey. Finally, the output layer emphasizes the main deliverables of the paper, namely the survey methodology, comparison with prior surveys, cross-domain synthesis of applications, identification of limitations and open gaps, and the future research roadmap.

This paper is organized as follows. Section Background of Generative AI presents the core concepts of GenAI and summarizes the main technical foundations needed for the rest of the paper. Sections GenAI in Healthcare–GenAI in Earth and Environment then review representative GenAI applications across major societal domains, including healthcare, agriculture, industry, transportation, and Earth/environmental systems, highlighting common tasks, benefits, and practical constraints. Section GenAI in Computer Science and Networks shifts to the computing perspective and discusses how GenAI is used within digital systems, including software and cybersecurity contexts. Sections GenAI in Materials Science and GenAI in Finance and Economics examine two data-intensive areas materials engineering and finance where GenAI supports discovery, prediction, and decision support under uncertainty. Sections GenAI in Education and Academia–GenAI in Arts and Creative Industries extend the discussion to education, business, and creative work, focusing on how GenAI changes workflows and human–AI interaction. Section Cross-Domain Synthesis, Risks, and Responsible Deployment synthesizes cross-domain trends and discusses risks, failure modes, ethics, fairness, privacy, and responsible deployment. Finally, Section Future Directions identifies open challenges and promising research directions, and Section Conclusion concludes the paper.

II. BACKGROUND OF GENERATIVE AI

Generative Artificial Intelligence, also known as Generative AI or GenAI, refers to a class of machine learning techniques that learn a probability distribution over complex data, such as images, text, audio, or molecular structures, and are able to sample from this distribution to produce novel, coherent instances. Instead of merely predicting labels or scalar outputs, generative models approximate the underlying data-generating process and thereby support synthesis, interpolation, editing, and conditional generation in a wide range of modalities. Building on advances in deep learning architectures, such as convolutional neural networks (CNNs) and long short-term memory (LSTM), recent progress in deep generative modeling, combined with large-scale datasets and accelerator hardware, has led to powerful foundation models that can generalize across tasks and domains, transforming areas such as vision, language, healthcare, and creative industries. In order to make the reviewed literature comparable, this section summarizes representative generative-model pipelines through high-level algorithmic workflows. These workflows are not intended to introduce new learning procedures; instead, they synthesize recurring methodological steps reported across different generative families. By expressing preprocessing, training, generation, validation, and application-oriented use in a unified notation, the survey provides a common reference structure for comparing methods that otherwise differ substantially in architecture, optimization, conditioning mechanisms, guidance strategies, regularization choices, and evaluation settings.

In general, three main steps are taken when using a generative model: Preprocessing, Training, and Generation. In the preprocessing step, the input data, such as the training dataset, are processed in a way that the model can use them for training. In the training step, the model learns how to generate output based on defined goals and metrics. Finally, in the generation step, the model generates the required output. There is an optional step following the generation step, referred to as Downstream Training. In this step, the synthetic data produced by the generative model are used to train a secondary, non-generative model, called the downstream model, designed to perform a specific, end-user application task, such as classification.

This general procedure is presented in Algorithm 1. $\mathcal{D}_{\text{raw}}$ is the raw dataset that is used for training the model. For example, it can be a dataset of images with their related captions. *is_conditional* is a boolean that indicates whether the model is conditional. A conditional model requires input data for generating the final result. For example, a conditional model might generate an image from a text

Algorithm 1 General GenAI Procedure.

Require: $\mathcal{D}_{\text{raw}}$ (Initial dataset of real data)
Require: *is_conditional* (State of being conditional)
Require: $\mathcal{I}$ (Required input for generation)
Require: g (Generative model family)
Require: N (Number of training iterations)
Ensure: $\mathcal{D}_{\text{syn}}, f_{\text{down}}$ (Synthetic data, Downstream model)
1: $\mathcal{D}_{\text{pre}} \leftarrow \varnothing$
2: **for each** $x \in \mathcal{D}_{\text{raw}}$ **do**
3: $\quad x \leftarrow \text{PreProcess}(x)$
4: $\quad \mathcal{D}_{\text{pre}} \leftarrow \mathcal{D}_{\text{pre}} \cup \{x\}$
5: Split $\mathcal{D}_{\text{pre}}$ into $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}, \mathcal{D}_{\text{test}}$
6: $\mathcal{G} \leftarrow \text{Initialize}(g)$
7: **for** $1 \leqslant i \leqslant N$ **do**
8: $\quad x \leftarrow$ Random member of $\mathcal{D}_{\text{train}}$
9: $\quad \tilde{x} \leftarrow \text{ForwardPass}(\mathcal{G}, x)$
10: $\quad loss \leftarrow \text{CalculateLoss}(x, \tilde{x})$
11: $\quad \mathcal{G} \leftarrow \text{Update}(\mathcal{G}, x, \tilde{x}, loss)$
12: $\mathcal{D}_{\text{syn}} \leftarrow \varnothing$
13: **if** *is_conditional* **then**
14: $\quad$ **for each** $i \in \mathcal{I}$ **do**
15: $\quad\quad \tilde{x} \leftarrow \text{ForwardPass}(\mathcal{G}, i)$
16: $\quad\quad$ **if** Validate$(\tilde{x}, i)$ **then**
17: $\quad\quad\quad \mathcal{D}_{\text{syn}} \leftarrow \mathcal{D}_{\text{syn}} \cup \{\tilde{x}\}$
18: **else**
19: $\quad$ **for** $1 \leqslant i \leqslant |\mathcal{I}|$ **do**
20: $\quad\quad \tilde{x} \leftarrow \text{ForwardPass}(\mathcal{G})$
21: $\quad\quad$ **if** Validate$(\tilde{x})$ **then**
22: $\quad\quad\quad \mathcal{D}_{\text{syn}} \leftarrow \mathcal{D}_{\text{syn}} \cup \{\tilde{x}\}$
23: $\mathcal{D}_{\text{aug}} \leftarrow \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{syn}}$
24: $f_{\text{base}} \leftarrow$ Initialize a baseline model
25: Train f_{base} on $\mathcal{D}_{\text{train}}$
26: $f_{\text{down}} \leftarrow$ Initialize a task-specific downstream model
27: Train f_{down} on $\mathcal{D}_{\text{aug}}$ with validation on $\mathcal{D}_{\text{val}}$
28: Evaluate f_{down} and f_{base} on $\mathcal{D}_{\text{test}}$ and compare performance
29: **return** $\mathcal{D}_{\text{syn}}, f_{\text{down}}$

caption, whereas an unconditional model might generate an image using only random noise. $\mathcal{I}$ is the required input for conditional models. For example if $\mathcal{I} = \{\text{label1}, \text{label2}, \text{label3}\}$, the algorithm generates three outputs related to label1, label2, and label3. If the model is unconditional, $\mathcal{I}$ is the total number of synthesizes the user requires. For example, if $\mathcal{I} = 5$ for an unconditional model, the model generates five synthesized items based on what it has learnt. g is the generative model family, which can be one of *Variational Autoencoders (VAEs)*, *Generative Adversarial Networks (GANs)*, *Diffusion models*, *Autoregressive models*, or *Flow-based / Normalizing Flow models*. These families are further explained in the remainder of this section. Finally, N is the total number of iterations for training the model. This algorithm has two outputs: $\mathcal{D}_{\text{syn}}$, which is the set of synthesized outputs generated by the trained model, and f_{down}, which is the trained downstream model, referring to any secondary predictive model that utilizes the learned representations from the generative model.

Lines 1 to 5 indicate the preprocessing step. Each of the entries in the raw dataset are preprocessed and then stored in a new set called $\mathcal{D}_{\text{pre}}$. The preprocess step is different for different applications and input types. For example, if the model works with images, the preprocessing step may contain normalizing and resizing of the images. For each application presented in this paper, we have explained the procedure of PreProcess(). Once the raw dataset is preprocessed, it is split into three subsets, $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{val}}$, and $\mathcal{D}_{\text{test}}$ for training, validating, and testing the model, respectively.

Lines 6 to 11 are dedicated to the training step. The generative model, $\mathcal{G}$, is first initialized based on its family type. The details of model initialization for each type are presented in Algorithm 2. Then, we have a loop that iterates N times for training $\mathcal{G}$. A random data, x, is selected from the training dataset and its related synthesized data is generated through ForwardPass(). The remainder of this section presents the details of ForwardPass() for each of the family types. The training loss is then calculated using CalculateLoss(). It is worth noting that there are two types of loss for generative models. One loss value is related to the generative model's family and it is calculated then in Update(). The other loss value is the loss specific to the application. The output of CalculateLoss() is the application loss. We have described CalculateLoss() for each of the applications we have presented in this paper. Once the application loss is calculated, it is passed to Update() where it is combined with the family-specific generative loss to update the parameters of $\mathcal{G}$. In the remainder of this section, Update() is described in details for each of the family types. In practice, the training step is typically performed in a batch-wise manner. However, for simplicity and clarity in the algorithm's presentation, a single-sample update is shown.

Lines 12 to 22 are for the generation step. An empty set, $\mathcal{D}_{\text{syn}}$ is defined for storing the synthesized data. For the case that the model is conditional (Line 13), for each of the input data in $\mathcal{I}$, an output is generated. ForwardPass() is called on the input data, and then, the output is checked to be valid using Validate(). If the synthesize data is valid, it is added to $\mathcal{D}_{\text{syn}}$. The validation process depends on the application of the generative model. Hence, for each of the applications presented in this paper, we have mentioned the details of Validate(). On the other hand, for the case that the model is unconditional (Line 18), the algorithm generates $\mathcal{I}$ different outputs within a loop. Again, ForwardPass() is called, but this time the input data is not required. The validation is also performed for this case.

Lines 23 to 28 are for the downstream training. First, the training set ($\mathcal{D}_{\text{train}}$) and the synthesized set ($\mathcal{D}_{\text{syn}}$) are augmented to form a bigger set ($\mathcal{D}_{\text{aug}}$). A base model (f_{base}) is initialized and trained on $\mathcal{D}_{\text{train}}$ to act as a baseline for evaluating the downstream model. The downstream model is trained on the augmented set and validated with $\mathcal{D}_{\text{val}}$. The performance of the downstream model can now be evaluated by comparing its results with the baseline model on the test set. In many practical scenarios, the downstream model may also reuse the encoder or latent representation learned by the generative model, but Algorithm 1 abstracts away such architectural sharing.

From a modeling perspective, GenAI encompasses several families of architectures that differ mainly in how they parameterize and learn the data distribution. Variational AutoEncoders (VAEs) learn explicit latent-variable models by maximizing a tractable variational lower bound on the log-likelihood. Generative Adversarial Networks (GANs) define implicit distributions through an adversarial game between a generator and a discriminator. Diffusion probabilistic models construct a Markov chain that gradually corrupts data with noise, and then learn a neural network to reverse this process, yielding high-fidelity samples with stable training dynamics. Autoregressive models factorize the joint distribution into a product of conditional terms, generating each element given its predecessors and forming the backbone of modern large language models. Finally, flow-based generative models apply invertible transformations to map a simple prior distribution to the data space, enabling exact likelihood computation and bidirectional mapping between data and latent variables. These families form the core building blocks for many contemporary generative systems, including text-to-image models, large language models, and multimodal foundation models.

The current GenAI landscape has also shifted toward foundation and multimodal systems. Large autoregressive language models such as GPT-style systems learn general-purpose next-token distributions over massive corpora and support instruction following, reasoning, coding, and text-based decision support [9]. Text-to-image and image-to-text systems extend this paradigm by combining language encoders, vision encoders, diffusion backbones, and alignment mechanisms; representative examples include early zero-shot text-to-image generation systems, latent diffusion models, and more recent large multimodal models [10, 11, 8]. Therefore,

TABLE 1. Comparison of this survey with representative prior GenAI surveys.

Survey	Main scope	Methodological coverage	Application coverage	Limitations	Position of this survey
Sengar et al. [5]	General GenAI taxonomy and applications	Broad model overview	General application domains	Limited domain-specific workflow analysis and limited synthesis of evaluation constraints	Provides a reviewed algorithmic abstraction and domain-aware interpretation of data, tasks, metrics, risks, and gaps
Gozalo-Brizuela and Garrido-Merchán [6]	Broad GenAI concepts and emerging applications	High-level taxonomy of generative approaches	Cross-domain overview	Less emphasis on reproducible survey scope and domain-specific critical comparison	Adds PRISMA-inspired methodology, comparison across domains, and explicit discussion of failure modes and responsible deployment
Zhang et al. [7]	Comprehensive GenAI review	Major families and selected applications	Broad but mainly descriptive coverage	Limited connection between model mechanisms, domain constraints, and deployment risks	Connects model families to implementation workflows, data structures, validation needs, and future research directions
Bommasani et al. [8]	Foundation models and societal implications	Foundation-model-centered analysis	Selected downstream areas	Focuses mainly on foundation models rather than the full spectrum of GenAI families and domain-specific applications	Covers both classical and modern GenAI families, including GANs, VAEs, diffusion, autoregressive, flow-based, foundation, and multimodal systems
This survey	Cross-domain GenAI applications and future directions	Reviewed unified workflow across VAE, GAN, diffusion, autoregressive, flow-based, foundation, and multimodal models	Healthcare, agriculture, industry, transportation, environment, networks, materials, finance, education, business, and creative industries	Broad scope prevents exhaustive treatment of every subdomain; selected representative studies are emphasized	Provides a transparent scope, comparison with prior surveys, critical synthesis, failure-mode discussion, ethical analysis, and domain-aware future roadmap

Algorithm 2 Initialize() Procedure in different families of GenAI

Require: g (GenAI model family)
Ensure: $\mathcal{G}$ (Initialized generative model)

if g = "VAE" **then**
  $E_\phi \leftarrow$ Initialize Encoder with its parameters ϕ
  $D_\theta \leftarrow$ Initialize Decoder with its parameters θ
  $\mathcal{G} \leftarrow$ Initialize model with E_ϕ and D_θ
else if g = "GAN" **then**
  $G_\phi \leftarrow$ Initialize Generator with its parameters ϕ
  $D_\theta \leftarrow$ Initialize Discriminator with its parameters θ
  $\mathcal{G} \leftarrow$ Initialize model with G_ϕ and D_θ
else if g = "Diff" **then**
  $N_\theta \leftarrow$ Initialize Neural Network with its parameters θ
  Set β as the predefined schedule
  Set T as the number of timesteps
  $\mathcal{G} \leftarrow$ Initialize model with N_θ, β, and T
else if g = "Autoreg" **then**
  $N_\theta \leftarrow$ Initialize Neural Network with its parameters θ
  Set S as the fixed output size
  $\mathcal{G} \leftarrow$ Initialize model with N_θ and S
else if g = "Flow" **then**
  $F_\theta \leftarrow$ Initialize invertible transformation with its parameters θ
  $F_\theta^{-1} \leftarrow$ Define inverse transformation analytically from F_θ
  $\mathcal{G} \leftarrow$ Initialize model with F_θ and F_θ^{-1}
return $\mathcal{G}$

the survey treats GANs, VAEs, diffusion, autoregressive, and flow-based models as foundational families, while recognizing that many deployed GenAI systems are hybrid foundation models that combine multiple components and training stages.

The Initialize() procedure for different families of GenAI is described in Algorithm 2.

A. VARIATIONAL AUTOENCODERS (VAES)

VAEs introduce a probabilistic formulation of the classical autoencoder, framing generation as inference in a latent-variable model [12]. Let x denote an observed data point and z a continuous latent variable. A VAE defines a Generative Model $p_\theta(x \mid z)$ and a Recognition Model $q_\phi(z \mid x)$ that approximates the intractable posterior $p_\theta(z \mid x)$. These models are also known as the Decoder and the Encoder, respectively.

This functionality can be clearly illustrated using the task of modeling and generating handwritten digits. Consider an observed image of a handwritten digit, denoted by x, which represents the observed data point. The VAE's objective is to map this image to a continuous latent variable z,a compact vector in the latent space, which encodes the fundamental style and structural characteristics of the digit, such as stroke thickness or slant. The VAE accomplishes this through two core probabilistic models. The first one is the recognition model or the encoder. Upon observing a specific image x, the encoder estimates the approximate posterior distribution of its corresponding latent code z. This distribution captures the most probable latent representations that could have generated x, pushing the encoding process beyond a simple point estimate. The second one is the generative model or the decoder. This model acts as a decoder that defines the likelihood of generating the original image x given a latent code z sampled from the approximated posterior. Furthermore, the latent space itself is regularized by a simple prior distribution $p(z)$, typically a standard Gaussian, ensuring that the learned features z are continuous and easily navigable for new sample generation. This probabilistic framing allows the VAE to not only reconstruct the input but also to model the underlying data distribution effectively.

The main objective in training a VAE is to maximize the marginal log-likelihood of the observed data, $\log p_\theta(x)$. However, direct computation of this term is computationally intractable due to the requirement of integrating over the continuous latent space. Consequently, VAE training relies on optimizing the Evidence Lower Bound (ELBO), which serves as a mathematically tractable lower bound on $\log p_\theta(x)$. Maximizing the ELBO ensures that the true marginal log-likelihood of the data is also driven upwards, thereby enabling parameter learning for both the generative and recognition models. The ELBO formulation is what allows the VAE to effectively balance data reconstruction fidelity and latent space regularization. This formulation, which defines the objective function for VAE training, is formally stated as:

$$\log p_\theta(x) \geqslant \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x \mid z)] - \mathrm{KL}\big(q_\phi(z \mid x) \parallel p(z)\big) \quad (1)$$

ELBO inequality strategically decomposes the intractable marginal

Algorithm 3 ForwardPass() Procedure in VAEs

Require: $\mathcal{G}$ (Generative model)
Require: x (Optional input data)
Ensure: $\tilde{x}$ (Synthesized output)
1: $E_\phi \leftarrow$ Encoder of $\mathcal{G}$
2: $D_\theta \leftarrow$ Decoder of $\mathcal{G}$
3: **if** x is provided **then**
4: $\quad \mu, \sigma \leftarrow E_\phi(x)$
5: $\quad \epsilon \leftarrow$ Sample from $\mathcal{N}(0, I)$
6: $\quad z \leftarrow \mu + \sigma \cdot \epsilon$
7: **else**
8: $\quad z \leftarrow$ Sample from $\mathcal{N}(0, I)$
9: $\tilde{x} \leftarrow D_\theta(z)$
10: **return** $\tilde{x}$

log-likelihood $\log p_\theta(x)$ into two principal, computationally tractable components that dictate the VAE's learning objective. The first major component of the ELBO is the Reconstruction Term, formally expressed as $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x \mid z)]$. The primary objective associated with this term is its maximization. This component serves as a quantitative measure of the reconstruction fidelity of the generative model, specifically, how accurately the decoder can recover the observed data x from its latent representation z. Operationally, this term computes the expected log-likelihood of the reconstructed output, averaged over the approximate posterior distribution $q_\phi(z \mid x)$. A higher value for the Reconstruction Term directly indicates that the decoder is highly successful at retrieving the input x from the latent codes proposed by the encoder. The second critical component of the ELBO is the Regularization Term $-\mathrm{KL}\big(q_\phi(z \mid x) \,\|\, p(z)\big)$. The effective objective is the minimization of the Kullback-Leibler (KL) divergence, $\mathrm{KL}\big(q_\phi(z \mid x) \,\|\, p(z)\big)$, which is achieved by maximizing the overall ELBO due to the negative sign in the formula. This term serves to measure the dissimilarity between the approximate posterior $q_\phi(z \mid x)$ and the simple prior distribution $p(z)$. Its core function is to enforce latent space regularization. By compelling the encoder to output posterior distributions that closely resemble the standard prior, the regularization term ensures that the latent space is continuous, smooth, and densely covered by the data representations. This structured organization is vital for the VAE's generative capacity, as it allows for meaningful and controlled synthesis of novel data points by simple sampling from the prior distribution $p(z)$.

To enable gradient-based optimization of the VAE's objective function, the Reparameterization Trick is essential. The challenge arises because the expectation operator ($\mathbb{E}$) in the reconstruction term involves stochastic sampling from the encoder's distribution, $q_\phi(z \mid x)$, which makes standard backpropagation impossible. The Reparameterization Trick addresses this by expressing the latent variable z as a deterministic function of the distribution's parameters and an external, unit-variance random variable (ϵ). Specifically, for a Gaussian approximate posterior, z is sampled as $z = \mu_\phi(x) + \sigma_\phi(x) \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. This transformation shifts the stochasticity outside the parameters ϕ and θ, allowing the gradient path to flow correctly through μ_ϕ and σ_ϕ. Consequently, this technique facilitates the calculation of low-variance gradient estimates, enabling the efficient training of the VAE's neural network components.

Considering the general GenAI algorithm presented in Algorithm 1, the forward pass process of VAE models is described in Algorithm 3. For the case that the model is conditional, the input x is provided. Hence, we have to generate the latent variable based on it (Lines 3 to 6). This part is done based on the Reparameterization Trick mentioned earlier. The encoder generates the mean and the standard deviation, and a sample variable is drawn from the corresponding Gaussian distribution. For the case that the second input is not provided, the latent variable is sampled directly from the prior distribution (Line 8). Finally, the latent variable is passed to the decoder and the synthesize output is generated (Line 9).

Algorithm 4 Update() Procedure in VAEs

Require: $\mathcal{G}$ (Generative model)
Require: x (Real data)
Require: $\tilde{x}$ (Synthesized data)
Require: l_{app} (Application-specific loss)
Ensure: $\mathcal{G}$ (Update generative model)
1: $l_{\text{recon}} \leftarrow \|x - \tilde{x}\|^2$
2: $E_\phi \leftarrow$ Encoder of $\mathcal{G}$
3: $\mu, \sigma \leftarrow E_\phi(x)$
4: $l_{\text{kl}} \leftarrow \sum_i \left(\mu_i^2 + \sigma_i^2 - \log(\sigma_i^2) - 1\right)/2$
5: $l_{\text{vae}} \leftarrow l_{\text{recon}} + l_{\text{kl}}$
6: $l_{\text{total}} \leftarrow l_{\text{vae}} + l_{\text{app}}$
7: $grads \leftarrow$ Backpropagate(l_{total})
8: Update ϕ and θ based on $grads$
9: Reassemble $\mathcal{G}$ with updated ϕ and θ
10: **return** $\mathcal{G}$

To update a VAE model, we use Algorithm 4. First, the reconstruction loss is calculated. The reconstruction loss in a VAE is typically defined as the negative log-likelihood of the observed data x given the latent variable z, approximated by $-\log p_\theta(x \mid z)$. However, calculating it directly can be computationally expensive, especially when the likelihood is hard to compute. To simplify, a commonly used assumption is that the data follows a Gaussian distribution with the mean and variance being the output of the decoder, which transforms the latent variable into the data space. In this algorithm, we have chosen to use the mean squared error between the original data and the reconstructed data as the reconstruction loss (Line 1). The next step is to calculate the KL divergence. In a VAE, the encoder outputs the mean $\mu \in \mathbb{R}^d$ and standard deviation $\sigma \in \mathbb{R}^d$ vectors, which parameterize a diagonal Gaussian posterior $q_\phi(z \mid x)$. Therefore, the KL divergence between $q_\phi(z \mid x)$ and the standard normal prior $p(z) = \mathcal{N}(0, I)$ can be computed in closed form as a sum over latent dimensions, as we have in Line 4. The ELBO-based loss is then formed as the sum of the reconstruction loss and the KL term (Line 5). This loss is combined with the application-specific loss to obtain the final training objective (Line 6). Finally, the gradients of all model parameters with respect to the total loss are computed using standard backpropagation (Line 7), and the encoder and decoder parameters are updated accordingly (Line 8).

VAEs offer a principled probabilistic framework, explicit likelihoods, and efficient amortized inference, making them attractive for representation learning, anomaly detection, and conditional generation. However, basic VAEs often produce samples that are blurrier than those from adversarial or diffusion-based models, due in part to the Gaussian output assumptions and the trade-off enforced by the ELBO between reconstruction fidelity and latent regularization. Numerous extensions, such as hierarchical VAEs, β-VAEs, and VAE–GAN hybrids, attempt to improve sample quality, disentanglement, or controllability while retaining the advantages of likelihood-based training.

B. GENERATIVE ADVERSARIAL NETWORKS (GANS)

GANs [13] formulate generative modeling as a two-player minimax game between a generator G and a discriminator D. The generator maps random noise $z \sim p(z)$, such as a simple Gaussian, to synthetic samples $G(z)$, while the discriminator attempts to distinguish real samples $x \sim p_{\text{data}}(x)$ from generated ones.

This functionality can be conceptualized as a continuous, two-player minimax game focused on synthesizing realistic data, such as human face images. The first player is the Generator, which aims to map a random noise vector z (typically sampled from a simple prior distribution, $z \sim p(z)$) into a synthetic data sample $G(z)$. The Generator's objective is purely deceptive: to produce samples that are indistinguishable from the real data. The second player is the Discriminator, which is trained simultaneously to differentiate

Algorithm 5 ForwardPass() Procedure in GANs

Require: $\mathcal{G}$ (Generative model)
Require: x (Optional input data)
Ensure: $\tilde{x}$ (Synthesized output)
1: $G_\phi \leftarrow$ Generator of $\mathcal{G}$
2: $z \leftarrow$ SampleFromNoiseDistribution()
3: **if** x is provided **then**
4: $\quad \tilde{x} \leftarrow G_\phi(z, x)$
5: **else**
6: $\quad \tilde{x} \leftarrow G_\phi(z)$
7: **return** $\tilde{x}$

between real samples $x \sim p_{\text{data}}(x)$ and synthetic samples $G(z)$, outputting a probability that the input is real. The adversarial learning process involves the Discriminator attempting to maximize its classification accuracy (maximizing the value function), while the Generator concurrently attempts to minimize this same value function by producing increasingly convincing samples. This dynamic zero-sum competition drives the models into an equilibrium where the generator creates highly realistic outputs, and the discriminator can no longer perform significantly better than random guessing (i.e., outputting 50%), signifying that the synthesized data closely approximate the true data distribution.

The original GAN objective is:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))] \quad (2)$$

The first phase of the minimax game focuses on the Discriminator, whose objective is to maximize the value function ($\max_D$). This maximization effort is achieved through two distinct, yet complementary, expected log-likelihood terms. The first term, $\mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)]$, incentivizes the Discriminator to correctly classify real data x by assigning high probability values, driving $D(x)$ toward 1. The second term, $\mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))]$, motivates D to correctly reject synthetic samples $G(z)$ by pushing the predicted probability $D(G(z))$ towards 0. Consequently, the optimization of the Discriminator forces it to become an expert classifier, simultaneously affirming the authenticity of samples from the true data distribution and denying the authenticity of generated outputs. The second critical component of the minimax objective involves the Generator, whose optimization goal is to minimize the overall value function ($\min_G$). The Generator's influence is restricted solely to the second term of the loss function; i.e., $\mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))]$, as it has no control over the first term concerning real data. The minimization strategy of the Generator is to produce synthetic samples, $G(z)$, that are highly convincing to the Discriminator, driving the predicted probability $D(G(z))$ towards 1. Achieving this causes the argument $\log(1 - D(G(z)))$ to approach $-\infty$, thereby successfully minimizing the objective. This adversarial pressure compels the Generator to evolve into a highly effective deceiver, resulting in the production of increasingly realistic and high-fidelity synthetic data. Under idealized conditions, this game converges to a Nash equilibrium where the generator recovers the true data distribution and the discriminator cannot improve beyond random guessing.

Based on the general GenAI algorithm mentioned in Algorithm 1, the forward pass process of GAN models can be shown as Algorithm 5. In GANs, the latent variable z is sampled from a user-defined prior noise distribution, typically a standard Gaussian $\mathcal{N}(0, I)$ or a uniform distribution $\mathcal{U}(-1, 1)$. Line 2 indicates this noise distribution. Unlike VAEs, which learn the posterior distribution, GANs never condition z on x. Therefore, sampling z is not dependent on x even if the model is conditioned. For the case the input data x is provided, or in other words, the model is conditioned (Line 3), the synthesized output is generated while x plays the role of conditioning variable. When the model is unconditioned, z is the only input passed to the generator (Line 6).

Algorithm 6 Update() Procedure in GANs

Require: $\mathcal{G}$ (Generative model)
Require: x (Real data)
Require: $\tilde{x}$ (Synthesized data)
Require: l_{app} (Application-specific loss)
Ensure: $\mathcal{G}$ (Update generative model)
1: $G_\phi \leftarrow$ Generator of $\mathcal{G}$
2: $D_\theta \leftarrow$ Discriminator of $\mathcal{G}$
3: $d \leftarrow D_\theta(x)$
4: $\tilde{d} \leftarrow D_\theta(\tilde{x})$
5: $l_{\text{d}} \leftarrow -\log(d) - \log(1 - \tilde{d})$
6: $grads_{\text{d}} \leftarrow$ Backpropagate(l_{d})
7: Update θ based on $grads_{\text{d}}$
8: $z \leftarrow$ SampleFromNoiseDistribution()
9: $\tilde{x}_{\text{new}} \leftarrow G_\phi(z)$
10: $\tilde{d}_{\text{new}} \leftarrow D_\theta(\tilde{x}_{\text{new}})$
11: $l_{\text{g}} \leftarrow -\log(\tilde{d}_{\text{new}}) + l_{\text{app}}$
12: $grads_{\text{g}} \leftarrow$ Backpropagate(l_{g})
13: Update ϕ based on $grads_{\text{g}}$
14: Reassemble $\mathcal{G}$ with updated ϕ and θ
15: **return** $\mathcal{G}$

The updating process of a GAN model is performed based on Algorithm 6. The discriminator is evaluated on both real data (x) and generated data ($\tilde{x}$), producing the scores $d = D_\theta(x)$ and $\tilde{d} = D_\theta(\tilde{x})$. The discriminator loss l_{d} is computed as the sum of two terms (Line 5): one encouraging the discriminator to assign high probability to real samples and another encouraging it to assign low probability to generated samples. The gradients of the discriminator parameters are then computed via standard backpropagation (Line 6), and the discriminator is updated accordingly (Line 7). Now, a new synthetic sample is first generated using the current generator parameters (Line 9), and the discriminator evaluates this sample to produce $\tilde{d}_{\text{new}}$ (Line 10). The generator loss is computed as the negative log-probability assigned by the discriminator to the generated sample being fake, encouraging the generator to produce samples classified as real. In addition, the application-specific loss is added to this term because such losses typically depend on the generated output and therefore propagate gradients only through the generator (Line 11). For simplicity, we assume in this generic formulation we assume that the application-specific loss is assigned solely to the generator objective, although in specific architectures it could, in principle, also be coupled to the discriminator. The total generator loss is then backpropagated to compute gradients with respect to the generator parameters (Line 12), and the generator is updated accordingly (Line 13).

GANs have been particularly successful in high-fidelity image synthesis, super-resolution, and style transfer, where sharpness and local realism are critical. They avoid explicit likelihood computation and instead rely on the discriminator to provide informative gradients that drive the generator toward the data manifold. At the same time, adversarial training is notoriously unstable and prone to issues such as mode collapse, where the generator learns to produce only a limited subset of the data distribution. Subsequent variants, such as Wasserstein GANs (WGAN), spectral normalization, and progressive growing, introduce alternative divergences, regularization schemes, and training curricula to stabilize optimization and expand support over the data distribution.

Conditional GANs (cGANs) extend the adversarial framework by incorporating side information y (e.g., class labels, text descriptions, semantic maps) into both the generator and discriminator. In a cGAN, the generator implements $G(z, y)$, producing samples that should be consistent with the condition y, while the discriminator is trained to distinguish real pairs (x, y) from synthetic pairs $\big(G(z, y), y\big)$. This simple modification enables controlled, conditional generation across modalities: image synthesis conditioned on labels or text, image-to-image translation between domains, or speech generation conditioned on speaker identity and content. cGANs thereby form a key component in many task-specific generative pipelines, supporting fine-grained control over attributes while benefiting from the sample quality of adversarial training. More advanced variants integrate attention mechanisms, multi-scale discriminators, or auxiliary classifiers to further enhance conditional fidelity and attribute disentanglement.

C. DIFFUSION PROBABILISTIC MODELS

Diffusion probabilistic models, also termed diffusion models or score-based models, are a newer class of generative models that have become state of the art in image and multimodal synthesis [14]. They are built around two stochastic processes: a forward diffusion process that gradually corrupts a data sample x_0 into nearly pure noise through a Markov chain $(x_t)_{t=0}^{T}$, and a learned reverse process that reconstructs data from noise.

Concretely, the forward process adds Gaussian noise with a predefined schedule $\{\beta_t\}_{t=1}^{T}$, and the reverse model is parameterized by a neural network that predicts either the original sample or the noise at each timestep. Sampling proceeds by starting from Gaussian noise and iteratively applying the learned denoising steps. Despite the apparent computational overhead of many diffusion steps, optimized schedules, improved samplers, and architectural advances (e.g., U-Net backbones, classifier-free guidance) enable diffusion models to achieve or surpass GAN-level visual fidelity, with more stable training and better mode coverage.

The Diffusion Probabilistic Model framework can be exemplified by the task of synthesizing high-fidelity, new images of natural scenes. The process hinges on two distinct mechanisms: the Forward Diffusion Process and the Learned Reverse Process. The Forward Process systematically and gradually corrupts an initial clean data sample x_0 (e.g., a clear image of a forest) into pure Gaussian noise over a predefined number of timesteps T (e.g., $T = 1000$). This is achieved via a Markov chain $(x_t)_{t=0}^{T}$, where at each step, a small, controlled amount of Gaussian noise is added according to a fixed variance schedule $\{\beta_t\}_{t=1}^{T}$. After T steps, the original image x_0 is transformed into a state x_T, which is essentially pure, featureless Gaussian noise. The core of the generative model lies in the Learned Reverse Process, which aims to invert this destruction. A neural network, is trained to predict and remove the exact Gaussian noise added at each timestep t. Alternatively, the network may predict the original sample x_0 or the mean of the reverse conditional distribution. To generate a new image, the process begins by sampling from pure Gaussian noise x_T and iteratively applying the learned denoising steps backward in time (from T to 0). This iterative denoising yields a complete, novel image x_0 upon reaching the final step. The key advantage of this gradual, step-by-step approach, as opposed to single-step generation, is its inherent stability during training and its proven capacity to achieve exceptional visual fidelity, particularly in high-resolution and complex synthesis tasks.

The forward pass process, shown as ForwardPass() in Algorithm 1, for diffusion models is presented in Algorithm 7. The algorithm begins by retrieving the neural denoising network, the predefined noise schedule, and the total number of timesteps (Lines 1 to 3). These components define the stochastic forward and reverse processes. When the conditioning input x is provided (Line 4), the algorithm first constructs its noisy counterpart by iteratively applying the forward diffusion process (Lines 6 to 8). At each timestep, a fresh Gaussian noise sample is drawn, and is applied using the forward diffusion rule parameterized by the noise schedule (β). This corresponds to producing noisy data under the predefined corruption process (Line 8). After the forward diffusion phase, the algorithm performs the reverse denoising process (Lines 9 to 11). For each timestep, in reverse order, the neural network predicts the noise component associated with the current state (Line 10). This prediction is used by the Denoise() function to update the sample according to the reverse-time dynamics, recovering a progressively cleaner estimate of the data. The final output is obtained from the fully denoised state (Line 12). In the unconditional case, generation is performed purely through reverse diffusion starting from noise (Line 14), producing the synthesized data after the final timestep (Lines 15 to 17).

The updating process for Diffusion models is presented in Algorithm 8. The algorithm begins by retrieving the denoising network, the noise schedule, and the total number of timesteps (Lines 1 to 3). A random timestep t and a Gaussian noise sample z are drawn (Lines 4 and 5). The diffusion scaling coefficients α and their cumulative product $\bar{\alpha}$ are then computed (Lines 6 and 7). Using these coefficients, a noised version of the real data x_{new} is constructed using the forward diffusion rule (Line 8). The network predicts the noise component associated with x_{new} (Line 9). The diffusion loss is defined as the mean squared error between the true noise z and the predicted noise $\tilde{z}$ (Line 10). This loss is combined with the application-specific loss to form the total objective (Line 11), and the model parameters are updated via standard backpropagation (Lines 12 to 14).

Algorithm 7 ForwardPass() Procedure in Diffusion Models

Require: $\mathcal{G}$ (Generative model)
Require: x (Optional input data)
Ensure: $\tilde{x}$ (Synthesized output)
1: $N_\theta \leftarrow$ Neural Network of $\mathcal{G}$
2: $\beta \leftarrow$ Predefined schedule of $\mathcal{G}$
3: $T \leftarrow$ Number of timesteps of $\mathcal{G}$
4: **if** x is provided **then**
5: $\quad y \leftarrow x$
6: $\quad$ **for** $1 \leqslant t \leqslant T$ **do**
7: $\quad\quad z \leftarrow$ Sample from $\mathcal{N}(0, I)$
8: $\quad\quad y \leftarrow$ ApplyNoise(y, z, t, β)
9: $\quad$ **for** $T \geqslant t \geqslant 1$ **do**
10: $\quad\quad \tilde{z} \leftarrow N_\theta(y, t)$
11: $\quad\quad y \leftarrow$ Denoise$(y, \tilde{z}, t, \beta)$
12: $\quad \tilde{x} \leftarrow y$
13: **else**
14: $\quad \tilde{x} \leftarrow$ Sample from $\mathcal{N}(0, I)$
15: $\quad$ **for** $T \geqslant t \geqslant 1$ **do**
16: $\quad\quad \tilde{z} \leftarrow N_\theta(\tilde{x}, t)$
17: $\quad\quad \tilde{x} \leftarrow$ Denoise$(\tilde{x}, \tilde{z}, t, \beta)$
18: **return** $\tilde{x}$

Algorithm 8 Update() Procedure in Diffusion Models

Require: $\mathcal{G}$ (Generative model)
Require: x (Real data)
Require: $\tilde{x}$ (Synthesized data, ignored for this model)
Require: l_{app} (Application-specific loss)
Ensure: $\mathcal{G}$ (Update generative model)
1: $N_\theta \leftarrow$ Neural Network of $\mathcal{G}$
2: $\beta \leftarrow$ Predefined schedule of $\mathcal{G}$
3: $T \leftarrow$ Number of timesteps of $\mathcal{G}$
4: $t \leftarrow$ Sample timestep from $\{1, \ldots, T\}$
5: $z \leftarrow$ Sample noise from $\mathcal{N}(0, I)$
6: $\alpha \leftarrow 1 - \beta$ ▷ element-wise
7: $\bar{\alpha} \leftarrow \prod_{i=1}^{t} \alpha_i$
8: $x_{\text{new}} \leftarrow x\sqrt{\bar{\alpha}} + z\sqrt{1 - \bar{\alpha}}$
9: $\tilde{z} \leftarrow N_\theta(x_{\text{new}}, t)$
10: $l_{\text{diff}} \leftarrow |z - \tilde{z}|^2$
11: $l_{\text{total}} \leftarrow l_{\text{diff}} + l_{\text{app}}$
12: $grads \leftarrow$ Backpropagate(l_{total})
13: Update θ based on $grads$
14: Reassemble $\mathcal{G}$ with updated θ
15: **return** $\mathcal{G}$

Algorithm 9 ForwardPass() Procedure in Autoregressive Models

Require: $\mathcal{G}$ (Generative model)
Require: x (Optional input data)
Ensure: $\tilde{x}$ (Synthesized output)
1: $N_\theta \leftarrow$ Neural Network of $\mathcal{G}$
2: $S \leftarrow$ Fixed output size of $\mathcal{G}$
3: **if** x is provided **then**
4: $\quad \tilde{x}_{\text{seq}} \leftarrow$ Tokenize(x)
5: **else**
6: $\quad \tilde{x}_{\text{seq}} \leftarrow$ An empty sequence
7: $\quad sos \leftarrow$ Predefined start of the sequence token of $\mathcal{G}$
8: $\quad$ Append sos to $\tilde{x}_{\text{seq}}$
9: **for** $1 \leqslant i \leqslant S$ **do**
10: $\quad tok \leftarrow$ Sample from $N_\theta(\tilde{x}_{\text{seq}})$
11: $\quad$ Append tok to $\tilde{x}_{\text{seq}}$
12: $\tilde{x}_{\text{seq}} \leftarrow$ The sequence of last S tokens in $\tilde{x}_{\text{seq}}$
13: $\tilde{x} \leftarrow$ Detokenize($\tilde{x}_{\text{seq}}$)
14: **return** $\tilde{x}$

Algorithm 10 Update() Procedure in Autoregressive Models

Require: $\mathcal{G}$ (Generative model)
Require: x (Real data)
Require: $\tilde{x}$ (Synthesized data, ignored for this model)
Require: l_{app} (Application-specific loss)
Ensure: $\mathcal{G}$ (Update generative model)
1: $N_\theta \leftarrow$ Neural Network of $\mathcal{G}$
2: $S \leftarrow$ Fixed output size of $\mathcal{G}$
3: $x_{\text{seq}} \leftarrow$ Tokenize(x)
4: $l_{\text{ar}} \leftarrow 0$
5: **for** $1 \leqslant i < S$ **do**
6: $\quad seq \leftarrow$ Subsequence of first i tokens in x_{seq}
7: $\quad p \leftarrow N_\theta(seq)$
8: $\quad l_{\text{ar}} \leftarrow l_{\text{ar}} - \log(p(x_{\text{seq}}[i+1]))$
9: $l_{\text{total}} \leftarrow l_{\text{ar}} + l_{\text{app}}$
10: $grads \leftarrow$ Backpropagate(l_{total})
11: Update θ based on $grads$
12: Reassemble $\mathcal{G}$ with updated θ
13: **return** $\mathcal{G}$

D. AUTOREGRESSIVE MODELS

Autoregressive generative models constitute one of the most fundamental classes of genAI architectures Tian et al. [15]. These models define a tractable factorization of the joint probability distribution over the components of a data sample. Given an observation $x = (x_1, x_2, \ldots, x_T)$, an autoregressive model decomposes the likelihood as:

$$p(x) = \prod_{t=1}^{T} p(x_t \mid x_{<t}) \tag{3}$$

where each conditional distribution is parameterized by a neural network.

The architecture of Autoregressive Models is most clearly understood through the example of sequential data generation, such as natural language processing. Considering an observed data sequence x, represented by the sentence "The sun is bright today," the model factorizes the joint probability $p(x)$ on its token $T = 5$ using the chain rule in Equation 3. This factorization dictates that the overall probability of observing the sequence is the product of conditional probabilities, where each token's likelihood is conditioned on the entire preceding history. For instance, the probability of the full sentence is decomposed as:

$$\begin{aligned} &p(\text{"The sun is bright today"}) = \\ &p(x_1) \times p(x_2 \mid x_1) \times p(x_3 \mid x_{1,2}) \times \ldots \times p(x_5 \mid x_{1,2,3,4}) \end{aligned} \tag{4}$$

During generation, the model proceeds sequentially (autoregressively): at each timestep t, the neural network parameterizes the conditional distribution $p(x_t \mid x_{<t})$, sampling the next token exclusively based on the sequence of tokens previously generated. This dependency on the complete historical context ensures that the generated output maintains coherence and contextual fidelity throughout the sequence.

The forward pass process for Autoregressive models is shown in Algorithm 9. First, the neural network and the output size (S) are retrieved from the model (Lines 1 and 2). A sequence, $\tilde{x}_{\text{seq}}$, is dedicated to the synthesized data. For the case where the model is conditional (Line 3), this sequence is the tokenized form of the input data (Line 4). For the unconditional case (Line 5), it is initiated with the predefined start of the sequence in $\mathcal{G}$ (Lines 6 to 8). Through a loop, this sequence is given to the neural network to predict the next token, and the predicted token is again appended to the sequence (Lines 9 to 11). It is worth noting that the output of the neural network (N_θ) is the distribution of tokens. Hence, to get the token, we can get a sample from it. Finally, the generated sequence is first cut to only have the last S tokens, and then detokenized to form the final output (Lines 12 and 13). It must be noted that in some scenarios, especially for text-based models, the generated output can have variable sizes. In this algorithm, for simplicity, we have assumed fixed-size outputs.

For updating an autoregressive model, we can follow Algorithm 10. In this algorithm, the input data is first tokenized (Line 3), and each of its tokens are investigated (Lines 5 to 8). In this loop, the distribution of the tokens is first predicted, and then the probability of the next token in this distribution is used for calculating the autoregressive loss (Line 8). Similar to other models, the application-specific loss is also added to form the final loss, and the model is updated based on it (Lines 9 to 12).

Early autoregressive models for images, such as PixelRNN and PixelCNN, demonstrated that strictly ordered, sequential prediction can produce high-fidelity samples while maintaining exact likelihood computation. More recently, the autoregressive principle has become the backbone of large-scale language and multimodal generation through Transformer architectures, which replace recurrence with self-attention, enabling parallelizable training and long-range contextual reasoning.

In the context of text generation, autoregressive Transformers such as GPT-3, GPT-4, PaLM, LLaMA, and their successors learn to predict the next token conditioned on an extremely large corpus of preceding text, allowing them to model complex syntax, semantics, factual knowledge, and reasoning behaviors. Their training objective typically minimizes the negative log-likelihood of the observed sequence, making them stable to train at scale and perfectly aligned with the autoregressive factorization. These models have since been extended to multimodal domains, such as image captioning, video generation, and code synthesis, by conditioning the sequence on embeddings from vision encoders, audio encoders, or graph neural networks.

Autoregressive models achieve strong performance thanks to their ability to model fine-grained dependencies and to leverage massive training corpora. They provide exact likelihood calculations, stable optimization characteristics, and a unified formulation across modalities. However, sampling speed can be a limitation, as generating a sequence requires evaluating the model sequentially step-by-step. Despite this, the autoregressive paradigm remains central in contemporary GenAI, particularly in language and multimodal applications, where its expressive power and scalability have proven unmatched.

E. FLOW-BASED GENERATIVE MODELS

Flow-based generative models represent another important family of deep generative architectures, characterized by their use of invertible neural transformations to map simple probability distributions into

Algorithm 11 ForwardPass() Procedure in Flow-Based Models

Require: $\mathcal{G}$ (Generative model)
Require: x (Optional input data)
Ensure: $\tilde{x}$ (Synthesized output)
1: $F_\theta^{-1} \leftarrow$ Inverse transformation of $\mathcal{G}$
2: **if** x is provided **then**
3: $\quad F_\theta \leftarrow$ Invertible transformation of $\mathcal{G}$
4: $\quad z \leftarrow F_\theta(x)$
5: **else**
6: $\quad z \leftarrow$ Sample from $\mathcal{N}(0, I)$
7: $\tilde{x} \leftarrow F_\theta^{-1}(z)$
8: **return** $\tilde{x}$

complex data distributions [16]. A flow model constructs a sequence of bijective mappings:

$$z = f(x), \quad x = f^{-1}(z) \tag{5}$$

where z follows a tractable prior distribution and f is an invertible function parameterized by a neural network. Because the transformation is invertible and differentiable, the exact data likelihood can be computed using the change-of-variables formula:

$$p(x) = p(f(x)) \left| \det \left(\frac{\partial f(x)}{\partial x} \right) \right| \tag{6}$$

This property allows flow-based models to perform exact maximum-likelihood training, something that most other generative architectures cannot do.

The functionality of Flow-Based Generative Models can be elucidated by considering the task of modeling a highly complex data distribution, such as that of realistic human face images. The architecture operates by defining an initial simple distribution $p(z)$, the prior, typically a standard Gaussian, in a latent space Z. The core mechanism involves a sequence of transformations implemented by a neural network, f, which defines a bijective mapping between the simple latent space z and the complex data space x. Crucially, because this transformation is both invertible and differentiable, the model can leverage the change-of-variables formula to compute the exact log-likelihood of any data sample x. This formula (Equation 6) requires the calculation of the determinant of the Jacobian matrix, $|\det(\frac{\partial f(x)}{\partial x})|$. This term acts as a volume correction factor, ensuring that the probability density is conserved as the simple distribution $p(z)$ is stretched and manipulated by the transformation f into the intricate data distribution $p(x)$. This unique property of being able to perform exact maximum-likelihood training is the primary computational advantage of flow-based models over other generative architectures like VAEs (which optimize a lower bound) and GANs (which do not possess an explicit likelihood function).

ForwardPass() for Flow-Based models can be defined as Algorithm 11. If the model is conditional (Line 2), the latent code is generated by passing the input data to the invertible transformation (Line 4), while in unconditional mode (Line 5), the latent code is sampled (Line 6). The synthesized output is generated by calling the inverse transformation on the latent code (Line 7).

The updating process for a Flow-Based model is shown in Algorithm 12. A latent code is generated using the invertible transformation (Line 2). The Jacobian determinant is also computed (Line 3). The likelihood of the sample is then computed exactly using the change-of-variables formula. The prior term penalizes unlikely latent codes, while the Jacobian term accounts for the volume change induced by the transformation (Lines 4 and 5). These contributions form the flow-based loss, which is combined with the application-specific loss to obtain the final objective (Line 6). Standard backpropagation is used to update the parameters of the invertible transformation (Lines 7 to 9).

Algorithm 12 Update() Procedure in Flow-Based Models

Require: $\mathcal{G}$ (Generative model)
Require: x (Real data)
Require: $\tilde{x}$ (Synthesized data, ignored for this model)
Require: l_{app} (Application-specific loss)
Ensure: $\mathcal{G}$ (Update generative model)
1: $F_\theta \leftarrow$ Invertible transformation of $\mathcal{G}$
2: $z \leftarrow F_\theta(x)$
3: $j \leftarrow$ Jacobian determinant of F_θ at x
4: $l_{\text{prior}} \leftarrow -\log p_{\mathcal{N}}(z)$
5: $l_{\text{jac}} \leftarrow -\log |j|$
6: $l_{\text{total}} \leftarrow l_{\text{prior}} + l_{\text{jac}} + l_{\text{app}}$
7: $grads \leftarrow$ Backpropagate(l_{total})
8: Update θ based on $grads$
9: Reassemble $\mathcal{G}$ with updated θ
10: **return** $\mathcal{G}$

Prominent examples of flow-based models include NICE, RealNVP, and Glow, which introduced coupling layers and architectural constraints that make the Jacobian determinant computationally manageable while maintaining expressive transformations. Glow, in particular, demonstrated competitive image-generation quality with efficient sampling, enabling applications such as image editing, style transfer, and latent-space interpolation. Beyond images, normalizing flows have been successfully applied in scientific domains, including molecular generation, protein modeling, and physics-informed modeling, due to their ability to model continuous, high-dimensional distributions while preserving exact density estimation.

Flow models offer several advantages, such as tractable likelihoods, stable likelihood-based optimization, and invertible mapping between data and latent spaces. These properties make them suitable for tasks requiring uncertainty quantification, interpretable latent variables, or exact probability evaluation. However, their architectural constraints, such as invertibility and the need for computationally efficient Jacobians, can limit representational flexibility compared to diffusion or autoregressive models. Nevertheless, flow-based methods remain a key component of the modern generative modeling landscape, especially in domains requiring both expressivity and probabilistic rigor.

F. FAILURE MODES AND EVALUATION CHALLENGES

A critical survey of GenAI must consider not only model capabilities but also recurring failure modes and evaluation weaknesses. GANs can suffer from training instability and mode collapse, where the generator covers only part of the target distribution despite producing visually plausible samples. VAEs may experience posterior collapse, especially when powerful decoders ignore latent variables, and they may generate overly smooth outputs when likelihood assumptions are restrictive. Diffusion models generally offer more stable training and strong sample diversity, but they can be computationally expensive during sampling and may reproduce biases or artifacts in the training data. Autoregressive models are highly expressive but can hallucinate unsupported facts, propagate exposure bias during long generation, and produce fluent yet incorrect outputs. Flow-based models provide exact likelihoods but require invertibility and tractable Jacobians, which can restrict architectural flexibility.

Evaluation remains equally challenging. Image metrics such as FID and Inception Score can be sensitive to feature extractors and may not reflect domain-specific validity, especially in medical, industrial, or scientific imaging. Text metrics such as BLEU and ROUGE can underrepresent factuality, reasoning quality, safety, and usefulness. In finance, transportation, and environmental science, distributional realism must be assessed through tail behavior, temporal dependence, physical constraints, and downstream decision utility rather than visual or linguistic plausibility alone. Therefore, robust GenAI evaluation should combine statistical fidelity, task utility, diversity, calibration, constraint satisfaction, privacy leakage testing, human or expert assessment, and domain-specific risk measures.

These limitations motivate the domain-level analysis in the following sections. In each area, the suitability of a generative model depends not only on sample quality but also on data availability, conditioning variables, validation requirements, safety constraints, and the cost of incorrect generation.

III. GENAI IN HEALTHCARE

GenAI is transforming clinical and therapeutic pipelines by addressing persistent data-centric constraints in healthcare, spanning high-dimensional medical images, structured patient trajectories, and multimodal clinical text. A central driver is mitigating scarcity and severe class imbalance in medical imaging through high-fidelity synthetic data that preserves diagnostically relevant anatomy and pathology, with recent progress extending to controllable, conditional generation and realistic 3D volumes for broader model training and pre-training [17, 18, 19, 20]. Beyond augmentation, GenAI is used for reconstruction, enhancement, and cross-modality translation to reduce acquisition burden, motion artifacts, and imaging limitations, while emphasizing structural fidelity via clinically informed losses and priors [21, 22, 23]. In parallel, synthetic EHR and longitudinal trajectory generation supports privacy-aware data sharing and cohort enrichment by capturing mixed discrete, continuous, and temporal dependencies, typically evaluated by statistical fidelity, downstream utility, and resistance to privacy attacks [24, 25, 26, 27, 28, 29, 30, 31]. Multimodal vision-language generation further automates clinical narratives and radiology reports from images, where clinical factuality and hallucination control are critical and domain-specific evaluation is preferred over generic NLP metrics [32, 33]. Finally, recent work explicitly incorporates privacy-preserving and fairness-aware constraints to satisfy regulatory compliance and ethical requirements, ensuring that synthetic data remains both useful and equitable for diagnostic and therapeutic decision-making [34, 35, 36]. The reviewed healthcare studies are summarized in Table 2.

To keep the formulation minimal while matching the general GenAI workflow in Algorithm 1, image-centric categories commonly use labeled images

$$\mathcal{D}_{\text{img}} = \{(x_{\text{image}}, y)\}, \tag{7}$$

where y denotes pathology, anatomy, or modality attributes used for conditional synthesis. For enhancement or translation, and for multimodal report generation, the common abstraction is paired inputs and targets

$$\mathcal{D}_{\text{pair}} = \{(x_{\text{in}}, x_{\text{tgt}})\}, \tag{8}$$

where x_{in} can be corrupted images or clinical images, and x_{tgt} represents the desired reconstruction, translated modality, or associated clinical report.

From a critical perspective, healthcare GenAI requires a stricter validation standard than general-purpose image or text generation. GAN-based augmentation can improve minority-class coverage, but synthetic lesions may introduce unrealistic artifacts if clinical constraints are weak. Diffusion models provide stronger diversity and local fidelity for imaging, but their outputs still require radiological validation and privacy leakage assessment. Autoregressive report-generation systems are useful for workflow support, yet hallucinated findings, missing abnormalities, and weak grounding remain major barriers to clinical deployment. Consequently, evaluation should combine image quality, diagnostic consistency, downstream utility, privacy risk, subgroup fairness, and expert review rather than relying only on generic visual or language metrics.

TABLE 2. GenAI in healthcare: compact summary of reviewed categories.

Cat.	Focus	Data	GenAI	Key refs.
H1	Image augmentation and synthesis	X-ray/CT/MRI (2D and 3D), labels or organ conditions	GAN/Diff/All	[17, 18, 19, 20]
H2	Reconstruction, enhancement, translation	Corrupted images, undersampled scans, paired modalities	GAN/All	[21, 22, 23]
H3	Synthetic EHR and trajectories	Mixed-type structured records and longitudinal encounters	GAN/VAE/All/Autoreg	[24, 25, 26, 27, 28, 29]
H4	Clinical narratives from images	Medical images paired with reports and notes	Autoreg	[32, 33]
H5	Privacy and fairness constraints	Health images, EHR, and text with ethical and compliance evaluation	All	[34, 35]
H6	Privacy-preserving synthesis	Sensitive images and health records with disclosure-risk focus	Diff	[36]
H7	Private health data synthesis	EHR and clinical text with explicit privacy guarantees	VAE/Autoreg	[30, 31]

IV. GENAI IN AGRICULTURE

GenAI in agriculture accelerates data-driven and resilient precision farming by directly addressing persistent data bottlenecks across the value chain, including scarcity, imbalance, noise, and missing observations. In practice, GenAI learns high-dimensional agricultural distributions and produces high-fidelity proxies that are hard to collect at scale in real fields, improving both model training and operational decision-making. The reviewed agriculture studies are summarized in Table 3.

In plant disease and pest detection (A1), the dominant challenge is that real-world field datasets are small, highly imbalanced (rare or early-stage cases), and affected by illumination variability and background clutter. Generative models, particularly GANs and diffusion-based pipelines, synthesize realistic diseased patterns while preserving anatomical structure, enabling targeted augmentation for specific classes or severity levels and supporting systematic validation under controlled attribute changes [37, 38, 39, 40, 41, 42, 43, 44, 45, 46]. In phenotyping and trait modeling (A2), conditional synthesis is used to vary phenology and trait attributes to expand coverage of growth stages and conditions that are expensive to measure, improving robustness of trait estimators [47]. For remote sensing and land-cover mapping (A3), generative translation and restoration help mitigate cloud and haze degradation, harmonize sensor or domain differences, and enhance mapping resolution so that downstream monitoring receives more consistent inputs [48, 49, 50, 51, 52]. In yield, growth, and agro-climatic time-series modeling (A4), generative models are mainly used for imputation and denoising of multivariate streams and for scenario generation that supports forecasting and risk-aware planning under climate variability [53, 54]. Additional agriculture vision synthesis studies (A5) focus on enrichment and robustness across diverse imaging settings [55, 56]. Finally, text-based advisory systems (A6) provide contextual decision support, where grounding and expert evaluation are essential to reduce hallucinations and avoid unsafe agronomic recommendations [57].

To align with the general GenAI workflow in Algorithm 1 while keeping the formulation minimal, image-centric categories (A1, A2) typically use labeled visual data:

$$\mathcal{D}_{\text{img}} = \{(x_{\text{image}}, y)\}, \tag{9}$$

where y denotes disease, severity, or trait attributes used for conditional synthesis. For remote sensing restoration or translation (A3) and time-series modeling (A4), a common abstraction is paired input and target data:

$$\mathcal{D}_{\text{pair}} = \{(x_{\text{in}}, x_{\text{tgt}})\}, \tag{10}$$

TABLE 3. GenAI in agriculture: compact summary of reviewed categories.

Cat.	Focus	Data	GenAI	Key refs.
A1	Plant disease and pest detection	Leaf or fruit images, labels, severity, lesion traits	GAN / Diff / Flow / All	[37, 38, 39, 40, 41, 42, 43, 44, 45, 46]
A2	Phenotyping and trait modeling	Images with phenology or trait attributes	GAN	[47]
A3	Remote sensing mapping and restoration	Satellite or UAV, cloudy or hazy scenes, SAR and optical, translation	GAN	[48, 49, 50, 51, 52]
A4	Yield, growth, and time-series modeling	Agro-climatic, microclimate, sensing, pest series, missing or noisy records	GAN/ Diff/ VAE/ All	[53, 54]
A5	Other agriculture vision synthesis	Agriculture images for enhancement and robustness (misc.)	GAN	[55, 56]
A6	Text advisory and decision support	Prompts and farm scenarios with expert-grounded responses	Autoreg	[57]

where x_{in} may be corrupted observations or incomplete sequences and x_{tgt} represents the desired restored, translated, or completed output.

Across agriculture studies, the main methodological tension is between visual realism and biological or agronomic validity. GANs remain common for leaf and pest image augmentation because they are efficient for small visual datasets, but diffusion models are increasingly attractive when controllable disease severity, lighting variation, and background diversity are required. For remote sensing, translation models must preserve geospatial alignment and spectral consistency, while advisory LLMs require grounding in local crop, soil, weather, and management constraints. Therefore, robust evaluation should move beyond classifier improvement and include disease-feature fidelity, agronomic plausibility, cross-season transfer, and expert agronomist assessment.

V. GENAI IN INDUSTRY AND MANUFACTURING

GenAI is reshaping industry and manufacturing across the product lifecycle, from concept design to operational intelligence and quality control, by learning high-dimensional engineering distributions over structural topologies, process time-series, and robotics or configuration spaces. A recurring benefit is converting expensive solver-driven optimization into fast inference, enabling rapid synthesis of high-performing designs and accelerating exploration of diverse candidate solutions under varying requirements [58, 59, 60, 61, 62]. Beyond performance-only generation, constraint-aware pipelines explicitly incorporate design-for-manufacturability and additive manufacturing limitations, so generated geometries remain buildable under real feature-size, overhang, tolerance, and defect constraints, reducing iteration between simulation and physical validation [63, 64]. In production, GenAI is widely used for resilience against data imbalance and missing labels in inspection and monitoring. For visual inspection, generative models synthesize rare defect patterns for augmentation and, importantly, learn the manifold of normal appearance to enable unsupervised or semi-supervised anomaly localization via reconstruction or domain translation errors, supporting zero-defect strategies without requiring exhaustive pixel-level annotations [65, 66, 67, 68]. For sensor and process monitoring, generative modeling captures multivariate temporal dependencies of normal operation and flags incipient faults as deviations from predicted or reconstructed traces, while also enabling synthesis of minority or degradation trajectories for predictive maintenance and schedule-aware decision support [69, 70, 71]. Finally, human-in-the-loop settings position GenAI as a co-creative assistant for design refinement, small-batch quality prediction, and interactive decision support, where interpretability and expert steering help align generated options with tacit constraints and shop-floor realities [72, 73]. The reviewed industry and manufacturing studies are summarized in Table 4.

TABLE 4. GenAI in industry and manufacturing: compact summary of reviewed categories.

Cat.	Focus	Data	GenAI	Key refs.
I1	Generative product and structural design	Designs and physics fields, boundary and load conditions, solver feedback	GAN/Diff/VAE	[58, 59, 60, 61, 62]
I2	Manufacturable and AM-constrained design	Geometry with manufacturability constraints, AM defect and tolerance signals	VAE/GAN	[63, 64]
I3	Visual inspection and surface defect detection	Industrial images, scarce defect labels, normal-only data for anomaly learning	GAN	[65, 66, 67]
I4	Sensor-based monitoring and fault detection	Multivariate time-series, process traces, imbalanced fault events	GAN/All	[69, 70, 71]
I8	Zero-defect anomaly frameworks	Normal-only images and process traces, anomaly via deviation from baseline	GAN	[68]
I5	Defect and anomaly learning variants	Inspection data with improved generation or attention mechanisms	GAN/VAE	
I6	Planning, path and robotics-oriented generation	Robot tasks, paths, and action or plan representations	GAN/Autoreg	
I7	Human-in-the-loop co-creative decision support	Expert feedback, small-batch quality data, interactive refinement	GAN/All	[72, 73]

To keep the formulation minimal while remaining consistent with the general GenAI workflow in Algorithm 1, design-centric categories commonly rely on paired requirements and outputs,

$$\mathcal{D}_{\text{pair}} = \{(x_{\text{in}}, x_{\text{tgt}})\}, \tag{11}$$

where x_{in} represents boundary conditions, loads, or constraints and x_{tgt} is a generated topology or geometry. Inspection and monitoring categories often use images or time-series with optional labels,

$$\mathcal{D}_{\text{obs}} = \{(x_{\text{obs}}, y)\}, \tag{12}$$

where y may be absent in unsupervised settings and detection is performed via deviations from the learned normal baseline.

In industry and manufacturing, the suitability of GenAI depends strongly on whether the target is design generation, defect synthesis, anomaly detection, or process monitoring. GANs are effective for visual defect augmentation and normal-manifold learning, but they may generate defects that are visually plausible yet physically impossible. VAEs and flow-based models are useful when interpretable latent spaces or likelihood-based anomaly scores are needed, while diffusion models are increasingly relevant for topology, geometry, and high-fidelity defect generation. The principal gap is the lack of closed-loop validation that connects generated designs or anomalies to physical simulation, manufacturability, process constraints, and shop-floor expert feedback.

VI. GENAI IN TRANSPORTATION AND MOBILITY SYSTEMS

GenAI is redefining intelligence and operational efficiency in transportation and mobility systems by addressing the stochastic, multi-modal nature of urban dynamics and the need for high-fidelity, continuous data streams. In traffic management, generative spatio-temporal modeling improves forecasting realism and robustness by learning the traffic-state manifold over road graphs, which reduces peak over-smoothing and supports principled recovery of missing or corrupted sensor measurements through learned imputation [74, 75, 76, 77, 78, 79, 80]. In autonomous driving, generative predictors move beyond point estimates to probabilistic scenario generation, producing diverse future trajectories under map, rule, and interaction constraints, which is essential for uncertainty-aware and safety-oriented planning [81, 82, 83, 84, 85, 86, 87]. GenAI also enables privacy-preserving mobility data synthesis for sharing and simulation, preserving spatio-temporal statistics while mitigating re-identification risk and supporting robust model training under limited real data availability [88, 89, 90, 91]. For operations and control, generative models augment rare or extreme conditions and stabilize decision pipelines by denoising, imputing, or producing compact control-oriented representations for adaptive signal control and downstream optimization [92]. Finally, LLM-based systems extend generative capability to unstructured planning and communication, supporting dynamic decision support in public transit and mobility services [93]. The reviewed transportation studies are summarized in Table 5.

To keep notation consistent with Algorithm 1 while remaining compact, the dominant supervision patterns are graph-conditioned traffic modeling and context-conditioned trajectory generation:

$$\mathcal{D}_{\text{traffic}} = \{(x_{\text{traffic}}, x_{\text{road}})\}, \qquad \mathcal{D}_{\text{traj}} = \{(x_{\text{context}}, x_{\text{future}})\}. \tag{13}$$

Transportation GenAI is characterized by safety-critical uncertainty. Trajectory and traffic generation must preserve multi-agent interaction, map constraints, traffic rules, and rare-event behavior. GAN-based trajectory models can produce diverse futures but may underrepresent low-probability hazardous cases; diffusion models are better suited to multimodal trajectory distributions but require efficient sampling for real-time planning. LLM-based transit assistants can improve communication and planning, yet must be grounded in current schedules, policies, and operational constraints. Evaluation therefore requires not only prediction error but also diversity, collision risk, rule compliance, calibration, privacy preservation, and operational usefulness.

VII. GENAI IN EARTH AND ENVIRONMENT

GenAI is transforming earth and environmental science by closing two long-standing gaps: computational infeasibility of high-fidelity simulators and sparsity or incompleteness of real observations. A dominant use is surrogate modeling, where generative emulators approximate expensive Earth System Models (ESMs), enabling rapid ensemble generation for uncertainty quantification, sensitivity analysis, and calibration that would otherwise require prohibitive compute [96, 97]. In parallel, GenAI supports inverse and reconstruction tasks that are ill-posed under sparse or noisy measurements: it learns subgrid-scale physics for uncertainty-aware downscaling and bias correction in climate fields, produces probabilistic nowcasts, and improves fidelity of extremes critical for local impacts and hydrology [98, 99, 100]. Beyond climate variables, generative spatio-temporal models reconstruct hidden environmental states such as air-quality and pollutant concentration maps from limited monitoring networks, providing exposure-ready fields and probabilistic forecasts for early warning and policy analysis [101, 102]. Finally, scenario synthesis is used both for frequent processes and for rare, high-impact events, delivering realistic ensembles for risk assessment and resilience planning. The reviewed studies are summarized in Table 6.

To align with Algorithm 1 while remaining compact, the dominant supervision patterns can be viewed as conditional generation from coarse-to-fine fields or sparse-to-dense observations:

$$\mathcal{D}_{\text{down}} = \{(x_{\text{coarse}}, x_{\text{fine}})\}, \qquad \mathcal{D}_{\text{inv}} = \{(x_{\text{obs}}, x_{\text{state}})\}. \tag{14}$$

For earth and environmental systems, generative outputs must be judged against physical consistency and extreme-event fidelity.

TABLE 5. GenAI in transportation and mobility systems: compact summary of reviewed categories.

Cat.	Focus	Data	GenAI	Key refs.
T1	Traffic state forecasting and imputation	Spatio-temporal speed or flow tensors, road graph topology, weather or streaming updates, missing-value masks	GAN/All	[74, 75, 76, 77, 78, 79, 80]
T2	Trajectory prediction and motion planning	Multi-agent context, map geometry, traffic signals, intent features, future trajectories with constraints	GAN/Diff	[81, 82, 83, 84, 85, 86, 87]
T3	Synthetic mobility data, trajectories, and demand	GPS or OD trajectories, ride-hailing requests, carsharing usage, privacy constraints and utility targets	GAN , Diff , Autoreg	[88, 89, 90, 91]
T4	Traffic control and operational decision-making	Control states for intersections, multi-modal sensing, corrupted or incomplete measurements, scenario augmentation for RL	GAN	[92]
T5	Perception and traffic-scene data synthesis	Traffic scenes, semantic representations, synthetic perception data for ITS models and simulation pipelines	GAN	[94, 95]
T6	LLM-based transit planning and decision support	Prompts, service policies, schedules, user queries, grounded responses for planning and communication	Autoreg	[93]

GANs and diffusion models are useful for downscaling and surrogate modeling, but generic image similarity metrics are insufficient because small spatial errors may correspond to large physical consequences. Diffusion models offer probabilistic ensembles and improved diversity, whereas GAN-based surrogates often provide faster inference. The key gap is integrating generative learning with conservation laws, uncertainty quantification, sensor design, and decision-relevant validation for hazards, climate extremes, and policy scenarios.

VIII. GENAI IN COMPUTER SCIENCE AND NETWORKS

GenAI is reshaping computer science and networking by shifting emphasis from purely discriminative learning to distribution modeling, where complex system behaviors can be synthesized, completed, or translated into actionable artifacts. Across security, operations, wireless, and network management, the common driver is the limited availability of representative measurements and labels, the high cost of collecting realistic traces, and the need to share data without leaking sensitive information. In security analytics, generative models mitigate severe class imbalance in IDS/NIDS datasets by synthesizing realistic flow records and, in more adversarial settings, producing evasive traffic to stress-test robustness [116, 117, 118, 119, 120, 121, 122]. For system logs and telemetry, GenAI learns normal operational dynamics and supports self-supervised anomaly detection via pseudo-anomaly generation and prediction-based deviation scoring . In wireless communications, measurement-driven channel generation provides fast, high-fidelity emulation of non-stationary propagation, preserving correlation and capacity-relevant statistics needed for beamforming and resource allocation when real CSI is scarce [123, 124, 125, 126]. In network management, traffic-matrix generation and completion exploit latent structure to recover missing measurements and create realistic scenarios for planning and optimization, increasingly with uncertainty-aware sampling [127, 128]. Finally, GenAI supports quality assurance and privacy-preserving data release by generating valid,

TABLE 6. GenAI in earth and environment: compact summary of reviewed categories.

Cat.	Focus	Data	GenAI	Key refs.
E1	Climate/weather downscaling and nowcasting	Coarse ESM or reanalysis fields, radar sequences, observational targets, extreme-event tails	GAN/Diff	[98, 99, 100]
E2	Earth system and climate emulation	Long-horizon global fields, monthly-to-daily conditioning, internal variability, calibration targets	GAN/Diff	[96, 97]
E3	Fast hazard and hydraulic surrogates	Flood extent or depth maps, rainfall-runoff drivers, topography, boundary conditions	GAN	[103, 104, 105, 106, 107]
E4	Atmospheric composition and air quality	Station networks, gridded ozone or PM fields, meteorology covariates, sparse-to-dense mapping	GAN	[101, 102]
E5	Land-cover and environmental mapping synthesis	Remote-sensing imagery, labels, domain shifts, scenario generation for monitoring	GAN/Diff	[108, 109, 110]
E6	Subsurface and geophysical inverse modeling	Seismic or geophysical observations, property fields, sparse sensors, reconstruction targets	GAN	[111, 112]
E7	Seismic ground motion and wavefield synthesis	Waveforms, site effects, broadband signals, scenario ensembles for hazard analysis	GAN	[113, 114, 115]

TABLE 7. GenAI in computer science and networks: compact summary of reviewed categories.

Cat.	Focus	Data	GenAI	Key refs.
C1	IDS/NIDS augmentation and adversarial traffic	Flow features with labels, minority attacks, protocol constraints, evasive samples for robustness tests	GAN	[116, 117, 118, 119, 120, 121]
C2	Log, telemetry, and anomaly analytics	Log templates or sequences, metrics streams, pseudo-anomalies, prediction and deviation scoring	GAN	
C3	Wireless channel and PHY emulation	CSI/CIR measurements, MIMO spatial correlation, mobility or blockage context, capacity-relevant statistics	GAN	[123, 124, 125, 126]
C4	Traffic matrices and network management	OD traffic matrices with missing entries, topology context, latent completion and scenario generation	GAN/Diff	[127, 128]
C5	QA, test generation, and privacy-aware synthesis	Valid input distributions, structured traces, synthetic release with privacy guarantees for sharing and benchmarking	Autoreg	[129, 130, 131, 132]
C6	Automated test data and privacy-preserving traces	Coverage-driven test data generation, synthetic trajectories with structure-aware constraints and privacy controls	GAN/Diff	[133, 134]

coverage-improving test inputs and synthetic traces that maintain utility while reducing disclosure risks [129, 130, 131, 132, 133, 134]. The reviewed studies are summarized in Table 7.

To remain consistent with Algorithm 1 while compact, the dominant supervision patterns can be viewed as conditional generation from context or labels:

$$\mathcal{D}_{\text{net}} = \{(x_{\text{feat}}, x_{\text{label}})\}, \qquad \mathcal{D}_{\text{ctx}} = \{(x_{\text{signal}}, x_{\text{context}})\}. \tag{15}$$

In computer science and networks, GenAI is valuable because realistic traces are expensive, privacy-sensitive, and often imbalanced. However, synthetic traffic and logs must preserve protocol semantics, temporal dependence, and attack behavior rather than only marginal feature distributions. GAN-based IDS augmentation can improve minority attack detection but may overfit benchmark artifacts. Diffusion and autoregressive models are promising for traffic matrices, logs, and code or test generation because they capture sequence and context, but they also introduce risks of unrealistic samples, privacy leakage, and adversarial misuse. Evaluation should include distributional fidelity, downstream detection utility, protocol validity, robustness under adaptive attacks, and privacy audits.

IX. GENAI IN MATERIALS SCIENCE

Materials discovery is constrained by a vast, high-dimensional space spanning composition, crystal structure, and microstructure, where exhaustive simulation and experimentation are infeasible. GenAI shifts the workflow from predominantly forward evaluation to scalable inverse design by learning probabilistic priors over synthesizable candidates and proposing new materials that satisfy target properties under feasibility constraints. At the atomic scale, generative models produce thermodynamically plausible lattices and atomic arrangements and are typically paired with physics screening (e.g., DFT) or stability-aware objectives to filter candidates and balance diversity with validity [135, 136, 137, 138, 139]. In composition space, GenAI learns distributions of viable multi-component systems (alloys, complexes) and generates candidate ratios under property conditioning, enabling multi-objective exploration beyond brute-force search [140, 141]. At the mesoscale, generative microstructure synthesis provides statistically faithful 3D morphologies for computational homogenization and property prediction, reducing the cost of generating representative volume elements and enabling data augmentation for scarce imaging regimes [142, 143, 144, 145, 146, 147]. Finally, emerging pipelines extend generation into the temporal and closed-loop domains by reconstructing unobserved intermediate states along transformation pathways and integrating LLMs/agents with specialized generators to connect literature mining, candidate proposal, and validation into self-driving discovery loops [148, 149].

The reviewed materials-science studies are summarized in Table 8.

Materials-science GenAI differs from generic content generation because validity is constrained by chemistry, crystallography, thermodynamics, kinetics, and synthesizability. VAEs and GANs are useful for exploring latent composition and microstructure spaces, while diffusion and autoregressive graph or sequence models are increasingly suitable for crystal and molecular generation because they can incorporate conditional property targets and structural constraints. A central limitation is that many generated candidates are only computationally plausible until validated by DFT, molecular simulation, or experiment. Therefore, future surveys and systems should distinguish sample novelty from physical stability, manufacturability, and experimentally feasible synthesis pathways.

X. GENAI IN FINANCE AND ECONOMICS

Finance and economics are dominated by non-Gaussian stylized facts (heavy tails, volatility clustering, regime shifts) that weaken classical parametric assumptions and limit the coverage of historical simulations. GenAI addresses this by learning high-dimensional,

TABLE 8. The reviewed models in Materials Science.

Cat.	Focus	Data	GenAI	Key refs.
M1	Crystal and inorganic structure generation	Crystal lattices/atomic graphs, stability/-physics constraints, screened candidates	GAN / VAE / Diff /Autoreg	[135, 136, 137, 150, 138, 139]
M2	Composition-space inverse design	Multi-component compositions (alloys/complexes), synthesizability constraints, literature-extracted signals	VAE / GAN / Autoreg	[140, 141]
M3	Property-conditioned inverse pipelines	Target properties, surrogate/physics validation data, multi-site or data-limited lab settings	Autoreg / GAN / Diff / Flow-based	[151, 152, 153, 154]
M4	Microstructure synthesis	2D/3D microstructure images/volumes, scarce microscopy datasets, defect/porosity/phase patterns	GAN	[142, 143, 144, 145, 146, 147]
M5	Dynamics and transformation pathways	Sparse time observations, intermediate-state reconstruction, kinetic trajectories for evolution modeling	Autoreg / Flow-based	[148]
M6	LLM-assisted autonomous discovery	Unstructured scientific text and structured materials data; constraints, conditioning, experiment planning in-the-loop	Autoreg / agentic	[149, 139]

TABLE 9. The reviewed models in Finance and Economics.

Cat.	Focus	Data	GenAI	Key refs.
F1	Synthetic market paths and stress scenarios	Return/price time-series (multivariate/irregular), stylized facts, stress-testing signals, privacy-preserving releases	GAN , VAE ,Diff	[155, 156, 157, 158]
F2	Price forecasting and algorithmic trading	Market time-series and text (news/disclosures/sentiment), regime shifts/noise, synthetic scenarios for robust training	GAN , Autoreg	[162, 163, 164, 165, 166]
F3	Portfolio optimization and risk management	Multi-asset scenarios, tail dependence/co-movements, VaR/ES estimation, extreme-but-plausible stress tests	GAN	[159]
F4	Derivative pricing and payoff-relevant generation	Option/derivative paths, simulation-heavy pricing/hedging data, path-dependent payoffs, constraint-aware samples	GAN / Flow-based	[160, 161]
F5	Interpretable latent factors and uncertainty-aware modeling	Latent state factors, uncertainty-calibrated scenarios, controllable sampling for exploration/what-if analyses	VAE / Diff / Autoreg	[167]
F6	Financial LLMs and generative assistants	Financial text and tables/numbers, mixed reasoning/evaluation, grounding/faithfulness signals for deployment	Autoreg	

non-linear joint distributions over prices, returns, volatilities, and covariates, enabling the synthesis of statistically consistent market paths for stress testing, privacy-preserving sharing, and data augmentation for forecasting and trading [155, 156, 157, 158]. Beyond path generation, generative modeling supports scenario-based portfolio construction and tail-risk estimation (e.g., VaR/ES) by producing multi-asset samples that preserve non-linear dependence and extreme co-movements [159]. GenAI is also used for derivative pricing and uncertainty-aware valuation by learning risk-neutral or payoff-relevant dynamics and accelerating simulation-heavy pipelines [160, 161]. In parallel, specialized Financial LLMs extend generation to unstructured and mixed-format inputs, turning disclosures, news, and numerical tables into structured rationales, forecasts, and decision support, while raising the bar for grounding and evaluation in safety-critical workflows . A summary of reviewed work is provided in Table 9.

The reviewed finance and economics studies are summarized in Table 9.

In finance and economics, generative realism must be assessed through stylized facts, dependence structure, and decision impact rather than visual or linguistic quality. GANs and VAEs can synthesize market paths for augmentation and stress testing, but they may fail under regime shifts or underestimate rare tail co-movements. Diffusion models provide flexible scenario generation and uncertainty-aware sampling, while autoregressive financial LLMs support text-based reasoning but require strict grounding to avoid unsupported investment or risk claims. Key gaps include out-of-distribution validation, temporal leakage prevention, calibration of extreme events, regulatory explainability, and the integration of synthetic scenarios with portfolio and risk-management objectives.

XI. GENAI IN EDUCATION AND ACADEMIA

Education faces a structural tension between personalization (individualized attention) and scalability (serving large, diverse cohorts). GenAI reduces this gap by acting as a high-capacity content and cognitive engine that can generate instructional materials and assessments, provide contextualized feedback and grading support, and enable adaptive conversational tutoring at scale. Current research increasingly goes beyond productivity claims and evaluates cognitive and pedagogical outcomes, including how generative assistance reshapes learning strategies in STEM and writing, how reliable and controllable assessment generation can be made, and how synthetic/augmented educational data should be validated under privacy, bias, and integrity constraints [168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185]. Table 10 summarizes the reviewed studies.

The reviewed studies in Education and Academia are summarized in Table 10.

Education and academia highlight the difference between generating useful content and producing pedagogically valid learning support. Autoregressive LLMs dominate this domain because most tasks involve explanation, dialogue, assessment generation, feedback, and writing assistance. However, fluency does not guarantee correctness, fairness, or learning gain. Synthetic educational records generated by GANs or VAEs may support privacy-aware analytics, but they must preserve learning trajectories and subgroup behavior. Evaluation should therefore include factual correctness, pedagogical alignment, student learning outcomes, bias across learner groups, academic-integrity risks, and transparency of AI involvement.

XII. GENAI IN BUSINESS AND SERVICES

The service economy is being reshaped by GenAI's ability to scale personalization and operationalize unstructured knowledge across customer-facing and internal workflows. In external functions, GenAI automates the synthesis of marketing creatives and copy at high volume, supports hyper-personalized messaging across fragmented touchpoints, and powers conversational agents that can deliver more consistent customer support with lower latency. In commerce, it improves product discoverability by generating and enriching catalog content (descriptions, attributes, categories, and review-derived insights) at marketplace scale. Internally, GenAI acts as a productivity multiplier for knowledge work by summarizing complex information and automating documentation, while dedicated generative tabular models help mitigate privacy and imbalance constraints by synthesizing customer records for safe analytics and improving minority-class coverage for tasks such as churn prediction. Finally, HR-facing applications highlight both automation

TABLE 10. The reviewed models in Education and Academia.

Cat.	Focus	Data	GenAI	Key refs.
D1	Generative content and assessment creation	Questions/exercises, passages/materials with controlled difficulty, teacher rubrics/feedback, multimodal learning artifacts	Autoreg/GAN	[168, 169, 170, 171]
D2	Adoption, governance, and classroom integration	Institutional policies, educator practices, deployment and safe-use guidelines, system-level integration evidence	Autoreg	[180, 181, 182, 183]
D3	AI tutors and conversational learning support	Tutor dialogues (hints/explanations/metacognitive prompts), guardrails and evaluation protocols, learning-gain signals	Autoreg	[172, 173, 174]
D4	Learning analytics and data augmentation	Student records and learning traces, missingness/imbalance patterns, privacy/fidelity constraints, predictive targets	GAN/VAE	[175, 176, 177]
D5	Programming and STEM education (code-centric GenAI)	Code submissions, debugging traces, IDE logs, instructor workflows; learning strategy and concept mastery outcomes	Autoreg	[178, 179]
D6	Learning outcomes, integrity, and student behavior	Controlled/field study data, performance and effort allocation measures, integrity incidents, policy/intervention signals	Autoreg	[184, 185]
D7	Academic productivity and scholarly writing support	Scholarly drafting/editing workflows, writing revisions, transparency/quality measures, research-process impacts	Autoreg	
D8	Cross-cutting: content and writing support	Shared artifacts across teaching and scholarship (assignments, rubrics, drafts), controllability and integrity safeguards	Autoreg	

potential and risk: LLMs can support screening and drafting, but empirical evidence shows they may reproduce or amplify historical bias, making evaluation and governance essential [186, 187, 188, 189, 190, 191, 192, 193, 194]. Table 11 summarizes the reviewed studies.

Papers using GenAI in Business and Services are summarized in Table 11.

Business and service applications are dominated by autoregressive and multimodal foundation models because customer interaction, marketing, documentation, and knowledge-work tasks are language-rich. GAN-based tabular synthesis remains useful for privacy-preserving CRM analytics and imbalance mitigation, but business value depends on whether generated records preserve actionable patterns without exposing individuals. Major limitations include hallucinated product or policy claims, brand inconsistency, unfair treatment in HR or customer segmentation, and weak causal evidence for productivity gains. Evaluation should include task completion, factuality, customer satisfaction, fairness, privacy, human oversight cost, and measurable operational impact.

XIII. GENAI IN ARTS AND CREATIVE INDUSTRIES

Creative industries face high cognitive load and cost in manual content production, which constrains scale, diversity, and iteration speed. GenAI is shifting workflows toward prompt- and constraint-driven synthesis and functioning as a creative co-pilot that democratizes access to advanced creation tools. Across modalities, it enables controllable visual generation (style, texture, brush-stroke behavior), supports music creation by balancing local realism (timbre) with long-horizon structure (harmony and form), and improves long-form narrative generation by preserving coherence and character/plot consistency. At the industry level, GenAI accelerates pre-production for

TABLE 11. The reviewed models in Business and Services domain.

Cat.	Focus	Data	GenAI	Key refs.
B1	Marketing and advertising	Ad creatives/copy variants, campaign assets, audience/engagement signals, A/B testing outcomes	GAN/Autoreg	[186]
B2	Customer service and customer experience	Support conversations, knowledge-base documents, satisfaction/resolution metrics, response-time logs	Autoreg	[187]
B3	E-commerce and online marketplaces	Product catalogs, reviews, SEO keywords, attribute fields, category labels and marketplace metadata	Autoreg	[188, 189, 190, 191]
B4	CRM and analytics	Customer records (churn/segmentation), imbalanced classes, privacy-preserving synthetic tabular data	GAN	[192]
B5	Human resources and recruitment	CVs/job posts, screening/ranking signals, fairness/bias audit datasets, hiring outcomes	Autoreg	[193]
B6	Knowledge-work productivity and documentation	Workplace documents, meeting notes, reports/summaries, organizational knowledge bases, productivity outcomes	Autoreg	[194]

film/animation, scales ideation in product and graphic design toward prototype-ready concepts, and enables procedurally generated and interactive content in games. A central research theme across these applications is controllability and evaluation of algorithmic creativity, ensuring outputs remain aligned with human intent, while additional work targets restoration and reinterpretation of cultural heritage assets through generative inpainting and reconstruction. The reviewed studies are summarized in Table 12.

The reviewed studies in the field of Arts and Creative Industries are summarized in Table 12.

In arts and creative industries, GenAI evaluation is less about a single notion of correctness and more about controllability, novelty, coherence, authorship, and alignment with human intent. Diffusion models currently dominate high-quality visual generation and design ideation, while autoregressive models remain central for narrative, dialogue, lyrics, code-like game assets, and structured creative planning. GANs still appear in style transfer, restoration, and specialized image synthesis, but their relative dominance has declined in open-ended visual generation. Open gaps include provenance tracking, copyright-sensitive training, long-horizon narrative consistency, controllable multimodal editing, and reliable human-centered evaluation of creativity.

XIV. CROSS-DOMAIN SYNTHESIS, RISKS, AND RESPONSIBLE DEPLOYMENT

The domain reviews reveal several cross-domain patterns. First, model choice is strongly shaped by data structure. GANs remain common in small or medium-sized visual datasets, especially for augmentation and anomaly-oriented synthesis. VAEs and flow-based models are useful when latent interpretability, likelihood estimation, or uncertainty quantification are important. Diffusion models increasingly dominate high-fidelity image, scientific-field, and trajectory generation because they provide stable training and strong diversity. Autoregressive and multimodal foundation models

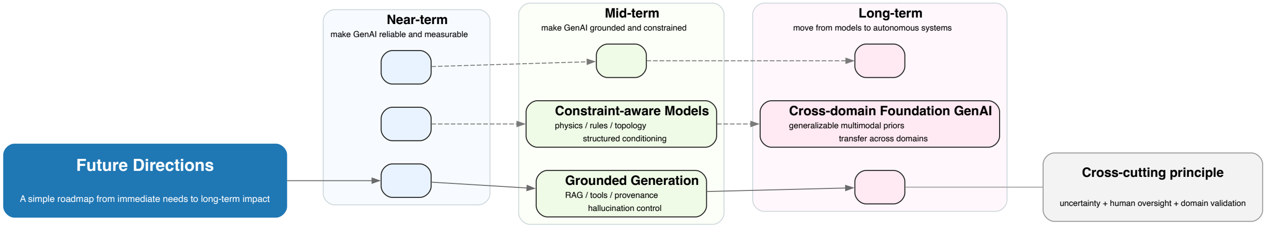

FIGURE 3. **Future directions roadmap of this survey, organized into near-term, mid-term, and long-term priorities, with a cross-cutting emphasis on uncertainty, human oversight, and domain validation.**

TABLE 12. **The reviewed models in Arts and Creative Industries domain.**

Cat.	Focus	Data	GenAI	Key refs.
R1	Visual arts and illustration	Images/artworks, style references, stroke/texture cues, composition controls, audience / creativity feedback	GAN / Diff	[195, 196, 197]
R2	Music composition and sound design	Symbolic music (MIDI / multi-track), raw audio, structure/timbre conditions, collaborative editing signals	VAE and Autoreg / Autoreg , All	[198]
R3	Narrative, story, and character generation	Text corpora (stories/scripts), character profiles, plot/structure constraints, interactive narrative traces	Autoreg	[199]
R4	Games and interactive content	Game levels/assets, gameplay constraints, play traces/evaluation signals, authoring-tool feedback	VAE / GAN / All	[200, 201]
R5	Product, graphic, and design ideation	Design concepts/sketches, prompts/briefs, prototype iterations, controllable design variations	Diff/All	[202, 203]
R6	Film/animation and story visualization	Storyboards/concept frames, scene/character consistency cues, sequence-level constraints, pre-production assets	GAN / Diff	
R7	Cultural heritage restoration and inpainting	Damaged artwork/images, masks/structure priors, restoration targets, preservation constraints	GAN/Diff	[204, 205]

dominate text-heavy, reasoning-heavy, and interaction-heavy settings such as education, business, clinical reporting, and creative writing. Second, evaluation must be domain-specific: the same synthetic sample may be visually plausible but clinically unsafe, statistically realistic but financially misleading, or fluent but factually unsupported. Third, the most useful GenAI systems are rarely pure generators; they are often embedded in closed-loop workflows that include retrieval, constraints, human feedback, simulation, safety filters, and downstream validation.

Responsible deployment is a central concern across the surveyed domains. In healthcare, finance, education, HR, and public services, generative systems may amplify bias, hallucinate harmful content, leak sensitive information, or create unjustified trust in synthetic outputs. Privacy-preserving generation must therefore be assessed through membership-inference, attribute-inference, and reconstruction-risk testing rather than assumed from visual dissimilarity. Fairness evaluation should consider subgroup performance, representational coverage, and possible harms caused by synthetic data imbalance. In language and multimodal systems, hallucination and weak grounding require retrieval support, source attribution, expert verification, and guardrails. Red-teaming and adversarial evaluation are also important for identifying harmful behaviors before deployment [206]. In scientific and engineering domains, generated outputs should be constrained by physical, chemical, regulatory, or operational rules.

A practical research agenda therefore requires four complementary safeguards: 1) transparent documentation of training data, inclusion criteria, and intended use; 2) domain-specific validation using expert knowledge and downstream decision metrics; 3) robustness testing under distribution shift, adversarial prompts, rare events, and missing data; and 4) governance mechanisms for human oversight, accountability, privacy, fairness, and provenance. These safeguards are especially important as GenAI shifts from offline synthesis toward real-time, multi-agent, and decision-support systems.

XV. FUTURE DIRECTIONS

The future of GenAI is increasingly shaped by three transitions: from isolated generators to grounded decision-support systems, from generic sample-quality metrics to domain-specific validation, and from single-modality models to multimodal foundation systems that combine language, vision, structured data, and simulation. These transitions are visible across all reviewed domains and directly motivate the roadmap below.

Building on the taxonomy presented in this survey, future GenAI research can be structured along two complementary axes: 1) the maturation of existing domain applications toward system-level, explainable, and causality-aware generative capabilities, and 2) the expansion into emerging or underexplored domains where GenAI remains largely nascent. Across domains, a unifying challenge is to move from isolated content or prediction generation toward closed-loop, policy-aware, and ethically grounded generative systems that operate under real-world constraints.

In **healthcare**, priority directions include causality- and physiology-constrained generative clinical co-pilots, trial emulation via virtual cohorts, and regulation-aware models that jointly optimize clinical outcomes, cost, and compliance, with particular emphasis on equitable deployment in low-resource settings. In **agriculture**, GenAI is expected to evolve toward multi-scale agro-ecosystem design by coupling generative models with crop, soil, and climate processes, enabling robust intervention portfolios, adaptive supply-chain contracts, and data gap filling via physically consistent synthetic sensing.

For **industry and manufacturing**, future work should shift toward closed-loop generative industrial ecosystems that integrate real-time telemetry, physics-based simulation, and supply-network co-design, including explainable failure scenario generation and privacy-preserving cross-factory knowledge transfer. In **transportation and mobility**, GenAI can advance toward multi-agent, policy-aware mobility ecosystems, supporting counterfactual urban design, safety-certified edge-deployable policies, and resilient planning under rare events.

In **earth and environmental sciences**, key directions include physically constrained generative earth-system experimentation, policy-aware environmental scenario generation, and generative design of sensing and field campaigns. In **computer science and networks**, GenAI may enable self-designing and self-explaining infrastructures through generative protocol synthesis, latent-space network operation, adversarial behavior generation, and standards evolution.

For **materials science**, future research should emphasize hierarchy-aware, manufacturing-constrained generation that co-designs materials, processes, and lifecycle strategies, supported by multi-fidelity discovery and autonomous laboratories. In **finance and economics**, GenAI can move toward generative policy and mechanism design, synthetic economies for stress testing, and inclusion-oriented financial products under fairness and regulatory constraints.

In **education and academia**, GenAI is expected to progress toward generative learning ecosystems, curriculum-level synthesis, and scientific co-creation, including generative peer review and institutional policy simulation. In **business and services**, future systems may support end-to-end generative organizations, encompassing business-model synthesis, adaptive service blueprints, and multi-agent organizational simulations.

In **arts, creative industries, and XR**, research directions include co-creative ecosystems, provenance-aware generation, living generative worlds, and context- and physiology-aware immersive experiences. Finally, high-impact but underexplored domains such as **public policy and governance**, **social protection and labour markets**, **energy systems and justice**, **humanitarian action**, and **cultural heritage** offer fertile ground for GenAI systems that co-design policies, institutions, and narratives under explicit ethical, legal, and societal constraints.

Figure 3 summarizes the forward-looking view of this survey as a three-horizon roadmap. Near-term priorities focus on measurable evaluation and data readiness, mid-term directions emphasize grounded and constraint-aware generation, and long-term opportunities move toward closed-loop, standardized, cross-domain GenAI systems aligned with real-world requirements.

XVI. CONCLUSION

Generative Artificial Intelligence (GenAI) has progressed from a collection of specialized modeling techniques to a foundational paradigm that shapes modern data-driven systems across domains. By learning rich probability distributions and enabling the synthesis, transformation, and simulation of complex data, GenAI has expanded the boundaries of what machine learning systems can generate and understand. However, the fast-growing and fragmented nature of existing literature has made it difficult for researchers and practitioners to identify unifying principles, compare domain-specific practices, or adapt generative techniques to emerging application settings.

This survey contributes to addressing these challenges by introducing a unified algorithmic framework that captures the core computational stages common to a wide range of generative models and by extending this framework to major families such as GANs, VAEs, diffusion models, autoregressive architectures, and flow-based approaches. Through a systematic and cross-domain examination of GenAI applications, spanning healthcare, agriculture, manufacturing, transportation, environmental science, computer networks, material science, finance, education, and business, we offer both a comprehensive overview of existing work and actionable algorithmic insights grounded in the unified framework. This dual perspective allows for a deeper understanding of how generative models are constructed, trained, and deployed across heterogeneous settings.

Looking ahead, the future directions outlined in this survey highlight new opportunities for advancing GenAI research, from domain-specific methodological improvements to entirely new areas where generative models have yet to be explored. As GenAI continues to mature, its impact will increasingly depend on interdisciplinary collaboration, robust evaluation practices, and careful alignment between generative capabilities and real-world constraints. The survey also shows that future progress should be measured not only by generative quality but also by robustness, factuality, privacy preservation, fairness, expert validation, and alignment with domain-specific constraints. We hope that the unified perspective presented in this survey provides a foundation for such progress and helps guide the next generation of GenAI innovations.

ACKNOWLEDGMENT

The work in this paper was supported in part by the Federal Ministry of Research, Technology, and Space (BMFTR), Germany, through the Project 6GEM+ under Grant 16KIS2411; and in part by the European Union through the 6G-Path project under Grant 101139172.

REFERENCES

[1] S. Yazdani, A. Singh, N. Saxena, Z. Wang, A. Palikhe, D. Pan, U. Pal, J. Yang, and W. Zhang, "Generative AI in depth: A survey of recent advances, model variants, and real-world applications," *Journal of Big Data*, vol. 12, no. 1, p. 230, 2025.

[2] A. Javadpour, F. Ja'fari, T. Taleb, M. Shojafar, and C. Benzaïd, "A comprehensive survey on cyber deception techniques to improve honeypot performance," *Computers & Security*, vol. 140, p. 103792, 2024.

[3] A. Javadpour, F. Ja'fari, T. Taleb, F. Turkmen, and C. Benzaïd, "Beyond reinforcement learning for network security: A comprehensive survey and tutorial," *Journal of Information Security and Applications*, vol. 96, p. 104294, 2026.

[4] S. A. Rabbani, M. El-Tanani, S. Sharma, S. S. Rabbani, Y. El-Tanani, R. Kumar, and M. Saini, "Generative artificial intelligence in healthcare: Applications, implementation challenges, and future directions," *BioMedInformatics*, vol. 5, no. 3, p. 37, 2025.

[5] S. S. Sengar, A. B. Hasan, S. Kumar, and F. Carroll, "Generative artificial intelligence: a systematic review and applications," *Multimedia Tools and Applications*, vol. 84, pp. 23 661–23 700, 2025.

[6] R. Gozalo-Brizuela and E. C. Garrido-Merchán, "A survey of generative AI applications," *Journal of Computer Science*, vol. 20, no. 8, pp. 801–818, 2024.

[7] Z. Zhang, J. Zhang, X. Zhang, and W. Mai, "A comprehensive overview of generative AI (GAI): Technologies, applications, and challenges," *Neurocomputing*, vol. 632, p. 129645, 2025.

[8] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[9] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1901.

[10] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 2021, pp. 8821–8831.

[11] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.

[12] A. Singh and T. Ogunfunmi, "An overview of variational autoencoders for source separation, finance, and bio-signal applications," *Entropy*, vol. 24, no. 1, p. 55, 2022.

[13] V. L. T. de Souza, B. A. D. Marques, H. C. Batagelo, and J. P. Gois, "A review on generative adversarial networks for image generation," *Computers & Graphics*, vol. 114, pp. 13–25, 2023.

[14] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 6840–6851.

[15] K. Tian, Y. Jiang, Z. Yuan, B. Peng, and L. Wang, "Visual autoregressive modeling: Scalable image generation via next-scale prediction," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 84 839–84 865.

[16] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel, "Flow++: Improving flow-based generative models with variational dequantization and architecture design," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 2019, pp. 2722–2730.

[17] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification," *Neurocomputing*, vol. 321, pp. 321–331, 2018.

[18] Y. Liu, L. Meng, and J. Zhong, "MAGAN: Mask attention generative adversarial network for liver tumor CT image synthesis," *Journal of Healthcare Engineering*, vol. 2021, p. 6675259, 2021.

[19] P. Guo, C. Zhao, D. Yang, Z. Xu, V. Nath, Y. Tang, B. Simon, M. Belue, S. Harmon, B. Turkbey *et al.*, "MAISI: Medical AI for synthetic imaging," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2025, pp. 4430–4441.

[20] F. Khader, G. Müller-Franzes, S. Tayebi Arasteh, T. Han, C. Haarburger, M. Schulze-Hagen, P. Schad, S. Engelhardt, B. Baeßler, S. Foersch *et al.*, "Denoising diffusion probabilistic models for 3d medical image generation," *Scientific Reports*, vol. 13, no. 1, p. 7303, 2023.

[21] K. Armanious, C. Jiang, S. Abdulatif, T. Küstner, S. Gatidis, and B. Yang, "Unsupervised medical image translation using cycle-MedGAN," in *2019 27th European Signal Processing Conference*. IEEE, 2019, pp. 1–5.

[22] K. Armanious, V. Kumar, S. Abdulatif, T. Hepp, S. Gatidis, and B. Yang, "ipA-MedGAN: Inpainting of arbitrary regions in medical imaging," in *2020 IEEE International Conference on Image Processing*. IEEE, 2020, pp. 3005–3009.

[23] H. Ali, M. R. Biswas, F. Mohsen, U. Shah, A. Alamgir, O. Mousa, and Z. Shah, "The role of generative adversarial networks in brain MRI: a scoping review," *Insights into Imaging*, vol. 13, no. 1, p. 98, 2022.

[24] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, "Generating multi-label discrete patient records using generative adversarial networks," in *Machine Learning for Healthcare Conference*. PMLR, 2017, pp. 286–305.

[25] M. K. Baowaly, C.-C. Lin, C.-L. Liu, and K.-T. Chen, "Synthesizing electronic health records using improved generative adversarial networks," *Journal of the American Medical Informatics Association*, vol. 26, no. 3, pp. 228–241, 2019.

[26] C. Sun, J. van Soest, and M. Dumontier, "Generating synthetic personal health data using conditional generative adversarial networks combining with differential privacy," *Journal of Biomedical Informatics*, vol. 143, p. 104404, 2023.

[27] S. Biswal, S. Ghosh, J. Duke, B. Malin, W. Stewart, C. Xiao, and J. Sun, "EVA: Generating longitudinal electronic health records using conditional variational autoencoders," in *Machine Learning for Healthcare Conference*. PMLR, 2021, pp. 260–282.

[28] J. Li, B. J. Cairns, J. Li, and T. Zhu, "Generating synthetic mixed-type longitudinal electronic health records for artificial intelligent applications," *NPJ Digital Medicine*, vol. 6, no. 1, p. 98, 2023.

[29] A. Warner, J. LeDue, Y. Cao, J. Tham, and T. H. Murphy, "Synthetic patient and interview transcript creator: an essential tool for LLMs in mental health," *Frontiers in Digital Health*, vol. 7, p. 1625444, 2025.

[30] J. Yoon, M. Mizrahi, N. F. Ghalaty, T. Jarvinen, A. S. Ravi, P. Brune, F. Kong, D. Anderson, G. Lee, A. Meir *et al.*, "EHR-Safe: generating high-fidelity and privacy-preserving synthetic electronic health records," *NPJ Digital Medicine*, vol. 6, no. 1, p. 141, 2023.

[31] M. M. Al Aziz, T. Ahmed, T. Faequa, X. Jiang, Y. Yao, and N. Mohammed, "Differentially private medical texts generation using generative neural networks," *ACM Transactions on Computing for Healthcare*, vol. 3, no. 1, pp. 1–27, 2021.

[32] Z. Chen, Y. Song, T.-H. Chang, and X. Wan, "Generating radiology reports via memory-driven transformer," *arXiv preprint arXiv:2010.16056*, 2020.

[33] J. Shentu and N. Al Moubayed, "CXR-IRGen: An integrated vision and language model for the generation of clinically accurate chest X-Ray image-report pairs," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 5212–5221.

[34] X. Zeng, F. Wang, Y. Luo, S.-g. Kang, J. Tang, F. C. Lightstone, E. F. Fang, W. Cornell, R. Nussinov, and F. Cheng, "Deep generative molecular design reshapes drug discovery," *Cell Reports Medicine*, vol. 3, no. 12, p. 100794, 2022.

[35] Y. Chen, Z. Wang, L. Wang, J. Wang, P. Li, D. Cao, X. Zeng, X. Ye, and T. Sakurai, "Deep generative model for drug design from protein target sequence," *Journal of Cheminformatics*, vol. 15, no. 1, p. 38, 2023.

[36] M. Tian, B. Chen, A. Guo, S. Jiang, and A. R. Zhang, "Reliable generation of privacy-preserving synthetic electronic health record time series via diffusion models," *Journal of the American Medical Informatics Association*, vol. 31, no. 11, pp. 2529–2539, 2024.
[37] M. Li, G. Zhou, A. Chen, J. Yi, C. Lu, M. He, and Y. Hu, "FWDGAN-based data augmentation for tomato leaf disease identification," *Computers and Electronics in Agriculture*, vol. 194, p. 106779, 2022.
[38] Y. Haruna, S. Qin, and M. J. Mbyamm Kiki, "An improved approach to detection of rice leaf disease with GAN-based data augmentation pipeline," *Applied Sciences*, vol. 13, no. 3, p. 1346, 2023.
[39] A. Benfenati, D. Bolzi, P. Causin, and R. Oberti, "A deep learning generative model approach for image synthesis of plant leaves," *PLOS ONE*, vol. 17, no. 11, p. e0276972, 2022.
[40] Q. H. Cap, H. Uga, S. Kagiwada, and H. Iyatomi, "LeafGAN: An effective data augmentation method for practical plant disease diagnosis," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 2, pp. 1258–1267, 2020.
[41] A. Muhammad, Z. Salman, K. Lee, and D. Han, "Harnessing the power of diffusion models for plant disease image augmentation," *Frontiers in Plant Science*, vol. 14, p. 1280496, 2023.
[42] N. Mori, H. Naito, and F. Hosoi, "Application of a latent diffusion model to plant disease detection by generating unseen class images," *AgriEngineering*, vol. 6, no. 4, pp. 4901–4910, 2024.
[43] I. Egusquiza, L. Benito-Del-Valle, A. Picón, A. Bereciartua-Pérez, L. Gómez-Zamanillo, A. Elola, E. Aramendi, R. Espejo, T. Eggers, C. Klukas *et al.*, "When synthetic plants get sick: Disease graded image datasets by novel regression-conditional diffusion models," *Computers and Electronics in Agriculture*, vol. 229, p. 109690, 2025.
[44] C. Wang, Y. Xia, L. Xia, Q. Wang, and L. Gu, "Dual discriminator GAN-based synthetic crop disease image generation for precise crop disease identification," *Plant Methods*, vol. 21, no. 1, p. 46, 2025.
[45] R. Zheng, J. Lee, H. Yin, D. Jin, and Y. H. Gu, "LITP-GAN: Leaf image translation pipeline using DETR and GLA-Net considering the lesion area of plant disease," *Journal of King Saud University – Computer and Information Sciences*, vol. 37, no. 4, p. 68, 2025.
[46] G. Han, D. K. P. Asiedu, and K. E. Bennin, "Plant disease detection with generative adversarial networks," *Heliyon*, vol. 11, no. 7, p. e43002, 2025.
[47] D. Pal and A. Ross, "Synthesizing forestry images conditioned on plant phenotype using a generative adversarial network," *Pattern Recognition*, vol. 162, p. 111385, 2025.
[48] L. E. Christovam, M. H. Shimabukuro, M. d. L. B. T. Galo, and E. Honkavaara, "Pix2Pix conditional generative adversarial network with MLP loss function for cloud removal in a cropland time series," *Remote Sensing*, vol. 14, no. 1, p. 144, 2022.
[49] C. Wang, K. Li, W. Yang, X. Wang, J. Wang, Z. Zhao, Y. Gu, and Z. Xu, "GACDNet: Mapping winter wheat by generative adversarial cross-domain networks with transformer integration for zero-sample extraction," *Computers and Electronics in Agriculture*, vol. 221, p. 109012, 2024.
[50] C. Shang, X. Li, G. M. Foody, Y. Du, and F. Ling, "Superresolution land cover mapping using a generative adversarial network," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2020.
[51] G.-H. Kwak and N.-W. Park, "Assessing the potential of multi-temporal conditional generative adversarial networks in SAR-to-optical image translation for early-stage crop monitoring," *Remote Sensing*, vol. 16, no. 7, p. 1199, 2024.
[52] J.-T. Ding, Y.-Y. Peng, M. Huang, and S.-J. Zhou, "AgriGAN: unpaired image dehazing via a cycle-consistent generative adversarial network for the agricultural plant phenotype," *Scientific Reports*, vol. 14, no. 1, p. 14994, 2024.
[53] C.-Y. Tai, W.-J. Wang, and Y.-M. Huang, "Using time-series generative adversarial networks to synthesize sensing data for pest incidence forecasting on sustainable agriculture," *Sustainability*, vol. 15, no. 10, p. 7834, 2023.
[54] I. S. Pothapragada and R. Sujatha, "GANs for data augmentation with stacked CNN models and XAI for interpretable maize yield prediction," *Smart Agricultural Technology*, p. 100992, 2025.
[55] X. Zhang, L. Yang, D. Zhang, T. Cui, X. He, K. Zhang, Z. Du, Z. Li, H. Li, and T. Xiao, "Smart agriculture technology: Real-time generation method of local soil property distribution maps based on WGAN-GPM," *Artificial Intelligence in Agriculture*, 2025.
[56] Y. Bao, X. Meng, H. Liu, M. Xu, and M. Wang, "A novel method for soil organic carbon prediction using integrated ground-air-space multimodal remote sensing data," *Geoderma*, vol. 460, p. 117453, 2025.
[57] R. Calone, E. Raparelli, S. Bajocco, E. Rossi, L. Crecco, D. Morelli, C. Bassi, R. Tiso, R. Bugiani, F. Pietrangeli *et al.*, "Analysing the potential of ChatGPT to support plant disease risk forecasting systems," *Smart Agricultural Technology*, vol. 10, p. 100824, 2025.
[58] S. Oh, Y. Jung, S. Kim, I. Lee, and N. Kang, "Deep generative design: Integration of topology optimization and generative models," *Journal of Mechanical Design*, vol. 141, no. 11, p. 111405, 2019.
[59] Z. Wang, S. Melkote, and D. W. Rosen, "Generative design by embedding topology optimization into conditional generative adversarial network," *Journal of Mechanical Design*, vol. 145, no. 11, p. 111702, 2023.
[60] F. Mazé and F. Ahmed, "Diffusion models beat GANs on topology optimization," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 8, pp. 9108–9116, 2023.
[61] Z. Nie, T. Lin, H. Jiang, and L. B. Kara, "TopologyGAN: Topology optimization using generative adversarial networks based on physical fields over the initial domain," *Journal of Mechanical Design*, vol. 143, no. 3, p. 031715, 2021.
[62] Y. Zhang, C. Jia, H. Zhang, N. Fang, S. Zhang, and N.-H. Kim, "Improving data-efficiency of deep generative model for fast design synthesis," *Journal of Mechanical Science and Technology*, vol. 38, no. 4, pp. 1957–1970, 2024.
[63] Z. Wang and H. Xu, "Manufacturability-aware deep generative design of 3D metamaterial units for additive manufacturing," *Structural and Multidisciplinary Optimization*, vol. 67, no. 2, p. 22, 2024.
[64] W. Chen, D. Lee, O. Balogun, and W. Chen, "GAN-DUF: Hierarchical deep generative models for design under free-form geometric uncertainty," *Journal of Mechanical Design*, vol. 145, no. 1, p. 011703, 2023.
[65] J. D. C. Mumbelli, G. A. Guarneri, Y. K. Lopes, D. Casanova, and M. Teixeira, "An application of generative adversarial networks to improve automatic inspection in automotive manufacturing," *Applied Soft Computing*, vol. 136, p. 110105, 2023.
[66] R. S. Peres, M. Azevedo, S. O. Araújo, M. Guedes, F. Miranda, and J. Barata, "Generative adversarial networks for data augmentation in structural adhesive inspection," *Applied Sciences*, vol. 11, no. 7, p. 3086, 2021.
[67] Y. Guo, L. Zhong, Y. Qiu, H. Wang, F. Gao, Z. Wen, and C. Zhan, "Using ISU-GAN for unsupervised small sample defect detection," *Scientific Reports*, vol. 12, no. 1, p. 11604, 2022.
[68] S. Ghansiyal, L. Yi, P. M. Simon, M. Klar, M. M. Müller, M. Glatt, and J. C. Aurich, "Anomaly detection towards zero defect manufacturing using generative adversarial networks," *Procedia CIRP*, vol. 120, pp. 1457–1462, 2023.
[69] J. E. Choi, D. H. Seol, C. Y. Kim, and S. J. Hong, "Generative adversarial network-based fault detection in semiconductor equipment with class-imbalanced data," *Sensors*, vol. 23, no. 4, p. 1889, 2023.
[70] J. Chung, B. Shen, and Z. J. Kong, "Anomaly detection in additive manufacturing processes using supervised classification with imbalanced sensor data based on generative adversarial network," *Journal of Intelligent Manufacturing*, vol. 35, no. 5, pp. 2387–2406, 2024.
[71] M. N. Ahsan, M. S. Islam, and M. M. Bappy, "Generative modeling in smart manufacturing: Applications, challenges, and future directions," *Manufacturing Letters*, vol. 44, pp. 1285–1295, 2025.
[72] A. Kusiak, "Convolutional and generative adversarial neural networks in manufacturing," *International Journal of Production Research*, vol. 58, no. 5, pp. 1594–1604, 2020.
[73] ——, "Generative artificial intelligence in smart manufacturing," *Journal of Intelligent Manufacturing*, vol. 36, no. 1, pp. 1–3, 2025.
[74] T. Sun, B. Sun, Z. Jiang, R. Hao, and J. Xie, "Traffic flow online prediction based on a generative adversarial network with multi-source data," *Sustainability*, vol. 13, no. 21, p. 12188, 2021.
[75] W. Sun, K. Deng, and J. Gao, "Traffic flow prediction based on generative adversarial network with hybrid spatio temporal features learning," *Cluster Computing*, vol. 28, no. 10, p. 621, 2025.
[76] D. Xu, Z. Lin, L. Zhou, H. Li, and B. Niu, "A GATs-GAN framework for road traffic states forecasting," *Transportmetrica B: Transport Dynamics*, vol. 10, no. 1, pp. 718–730, 2022.
[77] W. Zhang, R. Yao, Y. Yuan, X. Du, L. Wang, and F. Sun, "A traffic-weather generative adversarial network for traffic flow prediction for road networks under bad weather," *Engineering Applications of Artificial Intelligence*, vol. 137, p. 109125, 2024.
[78] W. Lin, Z. Zhang, Y. Zhao, J. Zhang, and G. Ren, "State-space and multi-scale convolutional generative adversarial network for traffic flow forecasting," *Systems*, vol. 13, no. 11, p. 991, 2025.
[79] J. Jin, D. Rong, T. Zhang, Q. Ji, H. Guo, Y. Lv, X. Ma, and F.-Y. Wang, "A GAN-based short-term link traffic prediction approach for urban road networks under a parallel learning framework," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 16 185–16 196, 2022.
[80] S. Choi, Z. Jin, S. W. Ham, J. Kim, and L. Sun, "A gentle introduction and tutorial on deep generative models in transportation research," *Transportation Research Part C: Emerging Technologies*, vol. 176, p. 105145, 2025.
[81] X. Li, G. Rosman, I. Gilitschenski, C.-I. Vasile, J. A. DeCastro, S. Karaman, and D. Rus, "Vehicle trajectory prediction using generative adversarial network with temporal logic syntax tree features," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3459–3466, 2021.
[82] H. Guo, Q. Meng, X. Zhao, J. Liu, D. Cao, and H. Chen, "Map-enhanced generative adversarial trajectory prediction method for automated vehicles," *Information Sciences*, vol. 622, pp. 1033–1049, 2023.
[83] C.-C. Hsu, L.-W. Kang, S.-Y. Chen, I.-S. Wang, C.-H. Hong, and C.-Y. Chang, "Deep learning-based vehicle trajectory prediction based on generative adversarial network for autonomous driving applications," *Multimedia Tools and Applications*, vol. 82, no. 7, pp. 10 763–10 780, 2023.
[84] C. Wei, G. Wu, M. J. Barth, A. Abdelraouf, R. Gupta, and K. Han, "Ki-GAN: Knowledge-informed generative adversarial networks for enhanced multi-vehicle trajectory forecasting at signalized intersections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7115–7124.
[85] H. Shi, S. Dong, Y. Wu, Q. Nie, Y. Zhou, and B. Ran, "Generative adversarial network for car following trajectory generation and anomaly detection," *Journal of Intelligent Transportation Systems*, vol. 29, no. 1, pp. 53–66, 2025.
[86] J. You, R. Gan, W. Tang, Z. Huang, J. Liu, Z. Jiang, H. Shi, K. Wu, K. Long, S. Fu *et al.*, "FollowGen: A scaled noise conditional diffusion model for car-following trajectory prediction," *Communications in Transportation Research*, vol. 5, p. 100215, 2025.
[87] Y. Liu, C. Zhu, X. Chang, X. Xi, C. Liu, and Y. Xu, "Diff-Pre: A diffusion framework for trajectory prediction," *Sensors*, vol. 25, no. 15, p. 4603, 2025.
[88] X. Ma, Z. Ding, and X. Zhang, "ST-TrajGAN: A synthetic trajectory generation algorithm for privacy preservation," *Future Generation Computer Systems*, vol. 161, pp. 226–238, 2024.
[89] H. Abbar, S. Kassan, F. Bidet, A. Cherigui, F. Guépin, and D. Renaud, "TrajDD-GAN: A synthetic mobility trajectory generation solution based on diffusion models," *IEEE Access*, 2025.
[90] T. Albrecht, R. Keller, D. Rebholz, and M. R"oglinger, "Fake it till you make it: Synthetic data for emerging carsharing programs," *Transportation Research Part D: Transport and Environment*, vol. 127, p. 104067, 2024.
[91] A. Jauhri, B. Stocks, J. H. Li, K. Yamada, and J. P. Shen, "Generating realistic ride-hailing datasets using GANs," *ACM Transactions on Spatial Algorithms and Systems*, vol. 6, no. 3, pp. 1–14, 2020.
[92] Z. Wang, H. Zhu, M. He, Y. Zhou, X. Luo, and N. Zhang, "GAN and multi-agent DRL based decentralized traffic light signal control," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1333–1348, 2021.
[93] S. Devunuri and L. Lehe, "TransitGPT: A generative AI-based framework for interacting with GTFS data using large language models," *Public Transport*, pp. 1–27, 2025.
[94] U. Jilani, M. Asif, M. Rashid, A. A. Siddique, S. M. U. Talha, and M. Aamir, "Traffic congestion classification using GAN-based synthetic data augmentation and a novel 5-layer convolutional neural network model," *Electronics*, vol. 11, no. 15, p. 2290, 2022.
[95] A. Ayub and H. Kim, "GAN-based data augmentation with vehicle color changes to train a vehicle detection CNN," *Electronics*, vol. 13, no. 7, p. 1231, 2024.
[96] S. Bassetti, B. Hutchinson, C. Tebaldi, and B. Kravitz, "DiffESM: Conditional emulation of temperature and precipitation in earth system models with 3d diffusion models," *Journal of Advances in Modeling Earth Systems*, vol. 16, no. 10, p. e2023MS004194, 2024.
[97] P. Hess, M. Aich, B. Pan, and N. Boers, "Fast, scale-adaptive and uncertainty-aware downscaling of earth system model fields with generative machine learning," *Nature Machine Intelligence*, vol. 7, pp. 363–373, 2025.
[98] N. Rampal, P. B. Gibson, S. Sherwood, G. Abramowitz, and S. Hobeichi, "A reliable generative adversarial network approach for climate downscaling and weather generation," *Journal of Advances in Modeling Earth Systems*, vol. 17, no. 1, p. e2024MS004668, 2025.
[99] F. Ling, Z. Lu, J.-J. Luo, L. Bai, S. K. Behera, D. Jin, B. Pan, H. Jiang, and T. Yamagata, "Diffusion model-based probabilistic downscaling for 180-year east asian climate reconstruction," *npj Climate and Atmospheric Science*, vol. 7, no. 1, p. 131, 2024.
[100] L. Glawion, J. Polz, H. Kunstmann, B. Fersch, and C. Chwala, "Global spatio-temporal ERA5 precipitation downscaling to km and sub-hourly scale using generative AI," *npj Climate and Atmospheric Science*, vol. 8, no. 1, p. 219, 2025.
[101] M. Cheng, F. Fang, I. M. Navon, J. Zheng, X. Tang, J. Zhu, and C. Pain, "Spatio-temporal hourly and daily ozone forecasting in china using a hybrid machine learning model: Autoencoder and generative adversarial networks," *Journal of Advances in Modeling Earth Systems*, vol. 14, no. 3, p. e2021MS002806, 2022.
[102] C. Yin, Y. Mao, L. Deng, M. Chen, Y. Rong, X. He, and X. Zhou, "STLLM-GAN: Spatio-temporal LLM generative adversarial network for PM2.5 prediction," *Expert Systems with Applications*, vol. 292, p. 128250, 2025.
[103] U. A. K. K. Perera, D. T. S. Coralage, I. U. Ekanayake, J. Alawatugoda, and D. P. P. Meddage, "A new frontier in streamflow modeling in ungauged basins with sparse data: A modified generative adversarial network with explainable AI," *Results in Engineering*, vol. 21, p. 101920, 2024.
[104] R. Sun, B. Pan, and Q. Duan, "Learning distributed parameters of land surface hydrologic models using a generative adversarial network," *Water Resources Research*, vol. 60, no. 7, p. e2024WR037380, 2024.
[105] J. Hofmann and H. Schüttrumpf, "FloodGAN: Using deep adversarial learning to predict pluvial flooding in real time," *Water*, vol. 13, no. 16, p. 2255, 2021.
[106] Z. Yin, Y. Saadati, B. Hu, A. S. Leon, M. H. Amini, and D. McDaniel, "Fast high-fidelity flood inundation map generation by super-resolution techniques," *Journal of Hydroinformatics*, vol. 26, no. 1, pp. 319–336, 2024.
[107] L. Huang, F. Antolini, A. Mostafavi, R. Blessing, M. Garcia, and S. D. Brody, "High-resolution flood probability mapping using generative machine learning with large-scale synthetic precipitation and inundation data," *Computer-Aided Civil and Infrastructure Engineering*, 2025.
[108] M. Gonzalez-Calabuig, M.-Á. Fernández-Torres, and G. Camps-Valls, "Generative networks for spatio-temporal gap filling of sentinel-2 reflectances," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 220, pp. 637–648, 2025.
[109] G. Li and G. Cao, "Generative adversarial models for extreme geospatial downscaling," *International Journal of Applied Earth Observation and Geoinformation*, vol. 139, p. 104541, 2025.
[110] G. Liu, H. He, Y. Gao, G. Zhang, and T. Cao, "Map geographic information road extraction method based on generative adversarial network and U-Net," *Scientific Reports*, vol. 15, no. 1, p. 25321, 2025.
[111] X. Jiang, X. Wang, Y. Liu, E. J. M. Carranza, S. Xie, and X. Wan, "Spatial extrapolation of downscaled geochemical data using conditional GAN," *Computers & Geosciences*, vol. 179, p. 105420, 2023.
[112] Z. Zhong, A. Y. Sun, and H. Jeong, "Predicting CO2 plume migration in heterogeneous formations using conditional deep convolutional generative adversarial network," *Water Resources Research*, vol. 55, no. 7, pp. 5830–5851, 2019.
[113] Y. Shi, G. Lavrentiadis, D. Asimaki, Z. E. Ross, and K. Azizzadenesheli, "Broadband ground-motion synthesis via generative adversarial neural operators: Development and validation," *Bulletin of the Seismological Society of America*, vol. 114, no. 4, pp. 2151–2171, 2024.
[114] M. Park, J. Bak, S. Park *et al.*, "Advanced wildfire detection using generative adversarial network-based augmented datasets and weakly supervised object localization," *International Journal of Applied Earth Observation and Geoinformation*, vol. 114, p. 103052, 2022.
[115] S. Khanmohammadi, M. G. Cruz, D. D. B. Perrakis, M. E. Alexander, and M. Arashpour, "Using AutoML and generative AI to predict the type of wildfire propagation in canadian conifer forests," *Ecological Informatics*, vol. 82, p. 102711, 2024.
[116] M. Ring, D. Schlör, D. Landes, and A. Hotho, "Flow-based network traffic generation using generative adversarial networks," *Computers & Security*, vol. 82, pp. 156–172, 2019.
[117] A. Ferriyan, A. H. Thamrin, K. Takeda, and J. Murai, "Generating network intrusion detection dataset based on real and encrypted synthetic attack traffic," *Applied Sciences*, vol. 11, no. 17, p. 7868, 2021.
[118] T. J. Anande, S. Al-Saadi, and M. S. Leeson, "Generative adversarial networks for network traffic feature generation," *International Journal of Computers and Applications*, vol. 45, no. 4, pp. 297–305, 2023.
[119] S. Rahman, S. Pal, S. Mittal, T. Chawla, and C. Karmakar, "SYN-GAN: A robust intrusion detection system using GAN-based synthetic data for IoT security," *Internet of Things*, vol. 26, p. 101212, 2024.
[120] Y. Yang, X. Liu, D. Wang, Q. Sui, C. Yang, H. Li, Y. Li, and T. Luan, "A CE-GAN based approach to address data imbalance in network intrusion detection systems," *Scientific Reports*, vol. 15, p. 7916, 2025.
[121] Z. Lin, Y. Shi, and Z. Xue, "IDSGAN: Generative adversarial networks for attack generation against intrusion detection," in *Advances in Knowledge Discovery and Data Mining*. Springer, 2019, pp. 217–237.
[122] A. Javadpour, P. Pinto, F. Ja'fari, and W. Zhang, "Dmaidps: A distributed multi-agent intrusion detection and prevention system for cloud iot environments," *Cluster Computing*, vol. 26, no. 1, pp. 367–384, 2023.
[123] Y. Yang, Y. Li, W. Zhang, F. Qin, P. Zhu, and C.-X. Wang, "Generative-adversarial-network-based wireless channel modeling: Challenges and opportunities," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 22–27, 2019.
[124] W. Xie, M. Xiong, Z. Yang, W. Liu, L. Fan, and J. Zou, "Real and fake channel: GAN-based wireless channel modeling and generating," *Physical Communication*, vol. 61, p. 102214, 2023.
[125] F. Euchner, J. Sanzi, M. Henninger, and S. ten Brink, "GAN-based massive MIMO channel model trained on measured data," in *2024 27th International Workshop on Smart Antennas*. IEEE, 2024, pp. 109–116.
[126] H. Cui, B. Xie, H. Wang, and V. C. M. Leung, "Generative-artificial-intelligence-based wireless channel modeling: Challenges and opportunities," *IEEE Communications Magazine*, vol. 63, no. 9, pp. 20–26, 2025.
[127] G. Kakkavas, M. Kalntis, V. Karyotis, and S. Papavassiliou, "Generative deep learning techniques for traffic matrix estimation from link load measurements," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 1029–1046, 2024.
[128] X. Yuan, Y. Qiao, Z. Wei, Z. Zhang, M. Li, P. Zhao, R. Hu, and W. Li, "Diffusion models meet network management: Improving traffic matrix analysis with diffusion-based approach," *IEEE Transactions on Network and Service Management*, vol. 22, no. 2, pp. 1259–1275, 2025.
[129] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman

et al., "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021.
[130] J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. Cai, M. Terry, Q. V. Le, and C. Sutton, "Program synthesis with large language models," *arXiv preprint arXiv:2108.07732*, 2021.
[131] Y. Li, D. Choi, J. Chung, N. Kushman, J. Schrittwieser, R. Leblond, T. Eccles, J. Keeling, F. Gimeno, A. Dal Lago *et al.*, "Competition-level code generation with AlphaCode," *Science*, vol. 378, no. 6624, pp. 1092–1097, 2022.
[132] N. Tao, A. Ventresque, V. Nallur, and T. Saber, "Enhancing program synthesis with large language models using many-objective grammar-guided genetic programming," *Algorithms*, vol. 17, no. 7, p. 287, 2024.
[133] X. Guo, H. Okamura, and T. Dohi, "Automated software test data generation with generative adversarial networks," *IEEE Access*, vol. 10, pp. 2190–2206, 2022.
[134] Y. Wang, G. Li, K. Li, and H. Yuan, "A deep generative model for trajectory modeling and utilization," *Proceedings of the VLDB Endowment*, vol. 16, no. 4, pp. 973–985, 2022.
[135] S. Kim, J. Noh, G. H. Gu, A. Aspuru-Guzik, and Y. Jung, "Generative adversarial networks for crystal structure prediction," *ACS Central Science*, vol. 6, no. 8, pp. 1412–1420, 2020.
[136] X. Luo, Z. Wang, P. Gao, J. Lv, Y. Wang, C. Chen, and Y. Ma, "Deep learning generative model for crystal structure prediction," *npj Computational Materials*, vol. 10, no. 1, p. 254, 2024.
[137] M. Alverson, S. G. Baird, R. Murdock, J. Johnson, T. D. Sparks *et al.*, "Generative adversarial networks and diffusion models in material discovery," *Digital Discovery*, vol. 3, no. 1, pp. 62–80, 2024.
[138] L. M. Antunes, K. T. Butler, and R. Grau-Crespo, "Crystal structure generation with autoregressive large language modeling," *Nature Communications*, vol. 15, no. 1, p. 10570, 2024.
[139] C. Zeni, R. Pinsler, D. Zügner, A. Fowler, M. Horton, X. Fu, Z. Wang, A. Shysheya, J. Crabbé, S. Ueda *et al.*, "A generative model for inorganic materials design," *Nature*, vol. 639, no. 8055, pp. 624–632, 2025.
[140] H. Turk, E. Landini, C. Kunkel, J. T. Margraf, and K. Reuter, "Assessing deep generative models in chemical composition space," *Chemistry of Materials*, vol. 34, no. 21, pp. 9455–9467, 2022.
[141] M. Strandgaard, T. Linjordet, H. Kneiding, A. L. Burnage, A. Nova, J. H. Jensen, and D. Balcells, "A deep generative model for the inverse design of transition metal ligands and complexes," *JACS Au*, vol. 5, no. 5, pp. 2294–2308, 2025.
[142] T. Hsu, W. K. Epting, H. Kim, H. W. Abernathy, G. A. Hackett, A. D. Rollett, P. A. Salvador, and E. A. Holm, "Microstructure generation via generative adversarial network for heterogeneous, topologically complex 3d materials," *JOM*, vol. 73, no. 1, pp. 90–102, 2021.
[143] A. Henkes and H. Wessels, "Three-dimensional microstructure generation using generative adversarial neural networks in the context of continuum micromechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 400, p. 115497, 2022.
[144] J.-W. Lee, N. H. Goo, W. B. Park, M. Pyo, and K.-S. Sohn, "Virtual microstructure design for steels using generative adversarial networks," *Engineering Reports*, vol. 3, no. 1, p. e12274, 2021.
[145] B. Murgas, J. Stickel, and S. Ghosh, "Generative adversarial network (GAN) enabled statistically equivalent virtual microstructures (SEVM) for modeling cold spray formed bimodal polycrystals," *npj Computational Materials*, vol. 10, no. 1, p. 32, 2024.
[146] S. Yamatoko, M. Kishimoto, S. Buchaniec, Y. Guo, and H. Iwai, "Advanced 3D microstructure generation of solid oxide cell electrodes using conditional generative adversarial network and validation using nonintuitive topological characteristics," *Advanced Intelligent Discovery*, vol. 1, no. 2, p. 202500010, 2025.
[147] R. Shams, M. Masihi, R. B. Boozarjomehry, and M. J. Blunt, "Coupled generative adversarial and auto-encoder neural networks to reconstruct three-dimensional multi-scale porous media," *Journal of Petroleum Science and Engineering*, vol. 186, p. 106794, 2020.
[148] D.-A. Dao, M.-Q. Ha, T.-S. Vu, S. Takazawa, N. Ishiguro, Y. Takahashi, M. Suzuki, T. Kakubo, N. Amino, H. Matsui *et al.*, "Material dynamics analysis with deep generative model," *Digital Discovery*, 2025.
[149] S. Lu, Q. Zhou, X. Chen, Z. Song, and J. Wang, "Inverse design with deep generative models: next step in materials discovery," *National Science Review*, vol. 9, no. 8, p. nwac111, 2022.
[150] F. Liu, Z. Chen, T. Liu, R. Song, Y. Lin, J. J. Turner, and C. Jia, "Self-supervised generative models for crystal structures," *iScience*, vol. 27, no. 9, p. 110672, 2024.
[151] C. Qin, J. Liu, S. Ma, J. Du, G. Jiang, and L. Zhao, "Inverse design of semiconductor materials with deep generative models," *Journal of Materials Chemistry A*, vol. 12, no. 34, pp. 22 689–22 702, 2024.
[152] J. Datta, A. Nadimpally, N. Koratkar, and D. Datta, "Generative AI for discovering porous oxide materials for next-generation energy storage," *Cell Reports Physical Science*, vol. 6, no. 7, p. 102665, 2025.
[153] D. Rajagopal, A. Cierpka, B. Nestler, and A. Koeppe, "Generative deep learning for advanced battery materials," *Batteries & Supercaps*, p. e202500494, 2025.
[154] C. Zhong, J. Zhang, X. Lu, K. Zhang, J. Liu, K. Hu, J. Chen, and X. Lin, "Deep generative model for inverse design of high-temperature superconductor compositions with predicted $T_c > 77$ K," *ACS Applied Materials & Interfaces*, vol. 15, no. 25, pp. 30 029–30 038, 2023.
[155] S.-I. Takahashi, M. Chen, and Y. Okuda, "Modeling financial time-series with generative adversarial networks," *Physica A: Statistical Mechanics and its Applications*, vol. 527, p. 121261, 2019.
[156] M. Wiese, R. Knobloch, R. Korn, and P. Kretschmer, "Quant GANs: Deep generation of financial time series," *Quantitative Finance*, vol. 20, no. 9, pp. 1419–1440, 2020.
[157] M. Dogariu, L.-D. Ştefan, B. A. Boteanu, C. Lamba, B. Kim, and B. Ionescu, "Generation of realistic synthetic financial time-series," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 18, no. 4, pp. 1–27, 2022.
[158] H. Huang, M. Chen, and X. Qiao, "Generative learning for financial time series with irregular and scale-invariant patterns," in *Proceedings of the 12th International Conference on Learning Representations*, 2024.
[159] J. H. Kim, S. Kim, Y. Lee, W. C. Kim, and F. J. Fabozzi, "Enhancing mean–variance portfolio optimization through GANs-based anomaly detection," *Annals of Operations Research*, vol. 346, no. 1, pp. 217–244, 2025.
[160] X. Tan, Z. Zhang, X. Zhao, and S. Wang, "DeepPricing: Pricing convertible bonds based on financial time-series generative adversarial networks," *Financial Innovation*, vol. 8, no. 1, pp. 1–38, 2022.
[161] H. G. Kim, S. J. Kwon, J. H. Kim, and J. Huh, "Pricing path-dependent exotic options with flow-based generative networks," *Applied Soft Computing*, vol. 124, p. 109049, 2022.
[162] Y. Lin, Z. Qin, B. Song, and J. Tang, "Stock price prediction based on generative adversarial network," *Applied Economics*, 2025.
[163] S. Li and S. Xu, "Enhancing stock price prediction using GANs and transformer-based attention mechanisms," *Empirical Economics*, vol. 68, no. 1, pp. 373–403, 2025.
[164] M. Vuletić, F. Prenzel, and M. Cucuringu, "Fin-GAN: Forecasting and classifying financial time series via generative adversarial networks," *Quantitative Finance*, vol. 24, no. 2, pp. 175–199, 2024.
[165] W. Kuryłek, "Can large language models forecast time series of earnings per share? case from poland," *Eastern European Economics*, pp. 1–18, 2025.
[166] G. Sperlì and M. A. Sichinolfi, "Empowered stock market forecasting using large language model on social media content," *Engineering Applications of Artificial Intelligence*, vol. 162, p. 112727, 2025.
[167] J. Wu, K. Plataniotis, L. Liu, E. Amjadian, and Y. Lawryshyn, "Interpretation for variational autoencoder used to generate financial synthetic tabular data," *Algorithms*, vol. 16, no. 2, 2023.
[168] S. Bhat, H. A. Nguyen, S. Moore, J. C. Stamper, M. Sakr, and E. Nyberg, "Towards automated generation and evaluation of questions in educational domains," in *Proceedings of the 15th International Conference on Educational Data Mining*, 2022.
[169] W. Morris, J. S. Choi, L. Holmes, V. Gupta, and S. Crossley, "Automatic question generation and constructed response scoring in intelligent texts," in *Proceedings of the 17th International Conference on Educational Data Mining*, 2024.
[170] L. Wang, R. Song, W. Guo, and H. Yang, "Exploring prompt patterns for generative artificial intelligence in automatic question generation," *Interactive Learning Environments*, vol. 33, no. 3, pp. 2559–2584, 2025.
[171] U. Bezirhan and M. von Davier, "Automated reading passage generation with OpenAI's large language model," *Computers and Education: Artificial Intelligence*, vol. 5, p. 100161, 2023.
[172] H. Bastani, O. Bastani, A. Sungu, H. Ge, Ö. Kabakcı, and R. Mariman, "Generative AI without guardrails can harm learning: Evidence from high school mathematics," *Proceedings of the National Academy of Sciences*, vol. 122, no. 26, p. e2422633122, 2025.
[173] Z. A. Pardos and S. Bhandari, "ChatGPT-generated help produces learning gains equivalent to human tutor-authored help on mathematics skills," *PLOS ONE*, vol. 19, no. 5, p. e0304013, 2024.
[174] G. Kestin, K. Miller, A. Klales, T. Milbourne, and G. Ponti, "AI tutoring outperforms in-class active learning: An RCT introducing a novel research-based design in an authentic educational setting," *Scientific Reports*, vol. 15, p. 17458, 2025.
[175] H. Farhood, I. Joudah, A. Beheshti, and S. Müller, "Advancing student outcome predictions through generative adversarial networks," *Computers and Education: Artificial Intelligence*, vol. 7, p. 100293, 2024.
[176] L. Zhang, J. Lin, J. Sabatini, C. Borchers, D. Weitekamp, M. Cao, J. Hollander, X. Hu, and A. C. Graesser, "Data augmentation for sparse multidimensional learning performance data in intelligent tutoring systems," *IEEE Transactions on Learning Technologies*, vol. 18, no. 1, pp. 145–162, 2025.
[177] S. Sarwat, N. Ullah, S. Sadiq, R. Saleem, M. Umer, A. A. Eshmawi, A. Mohamed, and I. Ashraf, "Predicting students' academic performance with conditional generative adversarial network and deep SVM," *Sensors*, vol. 22, no. 13, p. 4834, 2022.
[178] M. Lepp and J. Kaimre, "Does generative AI help in learning programming: Students' perceptions, reported use and relation to performance," *Computers in Human Behavior Reports*, vol. 18, p. 100642, 2025.
[179] D. Cambaz and X. Zhang, "Use of AI-driven code generation models in teaching and learning programming: A systematic literature review," in *Proceedings of the 55th ACM Technical Symposium on Computer Science Education*, 2024, pp. 172–178.
[180] J. S. Jauhiainen and A. Garagorry Guerra, "Generative AI in education: ChatGPT-4 in evaluating students' written responses," *Innovations in Education and Teaching International*, vol. 62, no. 4, pp. 1377–1394, 2025.
[181] ——, "Educational evaluation with large language models (LLMs): ChatGPT-4 in recalling and evaluating students' written responses," *Journal of Information Technology Education: Innovations in Practice*, vol. 24, p. Article 2, 2025.
[182] ——, "Generative AI in educational processes: ChatGPT-4 in providing feedback to students' written responses," *Research and Practice in Technology Enhanced Learning*, vol. 21, p. 27, 2026.
[183] J. Weidlich, F. Gotsch, K. Schudel, C. Marusic-Würscher, J. Mazzarella, H. Bolten, D. Bütler, S. Luger, B. Wohlfender, and K. Maag Merki, "Teacher, peer, or AI? comparing effects of feedback sources in higher education," *Computers and Education Open*, vol. 9, p. 100300, 2025.
[184] M. He, B. N. Abbasi, and J. He, "AI-driven language learning in higher education: An empirical study on self-reflection, creativity, anxiety, and emotional resilience in EFL learners," *Humanities and Social Sciences Communications*, vol. 12, no. 1, p. 1525, 2025.
[185] C. A. Cer'on Urz'ua, R. Ranjan, E. E. M'endez Saavedra, M. G. Badilla-Quintana, N. Lepe-Mart'inez, and A. Philominraj, "Effects of AI-assisted feedback via generative chat on academic writing in higher education students: A systematic review of the literature," *Education Sciences*, vol. 15, no. 10, p. 1396, 2025.
[186] K. Sohn, C. E. Sung, G. Koo, and O. Kwon, "Artificial intelligence in the fashion industry: Consumer responses to generative adversarial network (GAN) technology," *International Journal of Retail & Distribution Management*, vol. 49, no. 1, pp. 61–80, 2021.
[187] O. A. Abdelkader, "ChatGPT's influence on customer experience in digital marketing: Investigating the moderating roles," *Heliyon*, vol. 9, no. 8, p. e18770, 2023.
[188] A. Acharya, B. Singh, and N. Onoe, "LLM based generation of item-description for recommendation system," in *Proceedings of the 17th ACM Conference on Recommender Systems*, 2023.
[189] A. Qidwai, S. Mukhopadhyay, P. Khatiwada, D. Roth, and V. Gupta, "PRAISE: Enhancing product descriptions with LLM-driven structured insights," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2025, pp. 644–652.
[190] Z. Cheng, W. Zhang, C.-C. Chou, Y.-Y. Jau, A. Pathak, P. Gao, and U. Batur, "E-commerce product categorization with LLM-based dual-expert classification paradigm," in *Proceedings of the 1st Workshop on Customizable NLP*, 2024, pp. 294–304.
[191] K. I. Roumeliotis, N. D. Tselikas, and D. K. Nasiopoulos, "LLMs for product classification in e-commerce: A zero-shot comparative study of GPT and Claude models," *Natural Language Processing Journal*, vol. 11, p. 100142, 2025.
[192] I. N. M. Adiputra and P. Wanchai, "CTGAN-ENN: A tabular GAN-based hybrid sampling method for imbalanced and overlapped data in customer churn prediction," *Journal of Big Data*, vol. 11, no. 1, p. 121, 2024.
[193] S. S. Sivakaminathan and E. Musi, "ChatGPT is a gender bias echo-chamber in HR recruitment: An NLP analysis and framework to uncover the language roots of bias," *AI & Society*, pp. 1–21, 2025.
[194] M.-N. Chu, "Assessing the benefits of ChatGPT for business: An empirical study on organizational performance," *IEEE Access*, vol. 11, pp. 76 427–76 436, 2023.
[195] X. Gao, Y. Tian, and Z. Qi, "RPD-GAN: Learning to draw realistic paintings with generative adversarial network," *IEEE Transactions on Image Processing*, vol. 29, pp. 8706–8720, 2020.
[196] J. Xiang, "On generated artistic styles: Image generation experiments with GAN algorithms," *Visual Informatics*, vol. 7, no. 4, pp. 36–40, 2023.
[197] Q. Wang, C. Guo, H.-N. Dai, and P. Li, "Stroke-GAN painter: Learning to paint artworks using stroke-style generative adversarial networks," *Computational Visual Media*, vol. 9, no. 4, pp. 787–806, 2023.
[198] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.
[199] X. Peng, K. Xie, A. Alabdulkarim, H. Kayam, S. Dani, and M. Riedl, "Guiding neural story generation with reader models," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022, pp. 7087–7111.
[200] T. Tanabe, K. Fukuchi, J. Sakuma, and Y. Akimoto, "Level generation for angry birds with sequential VAE and latent variable evolution," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2021, pp. 1052–1060.
[201] A. Hald, J. S. Hansen, J. Kristensen, and P. Burelli, "Procedural content generation of puzzle games using conditional generative adversarial networks," in *Proceedings of the 15th International Conference on the Foundations of Digital Games*, 2020, pp. 1–9.
[202] J. Lee, J. Kim, and H. Lee, "DiffuseDesigner: Sketch-based controllable clothing image generation," *Fashion and Textiles*, vol. 12, no. 1, p. 26, 2025.
[203] J. X. Wu and L. Li, "AI-driven computational creativity in fashion design: A review," *Textile Research Journal*, vol. 95, no. 5–6, pp. 658–675, 2025.
[204] F. Zhao, H. Ren, K. Sun, and X. Zhu, "GAN-based heterogeneous network for ancient mural restoration," *Heritage Science*, vol. 12, no. 1, p. 418, 2024.
[205] G. Sumathi and M. Uma Devi, "Inpainting of damaged temple murals using edge- and line-guided diffusion patch GAN," *Frontiers in Artificial Intelligence*, vol. 7, p. 1453847, 2024.
[206] D. Ganguli, L. Lovitt, J. Kernion, A. Askell, Y. Bai, S. Kadavath, B. Mann, E. Perez, N. Schiefer, K. Ndousse *et al.*, "Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned," *arXiv preprint arXiv:2209.07858*, 2022.

AMIR JAVADPOUR Amir Javadpour is a Senior Cybersecurity Researcher MOSA!C Lab / ICTFICIAL Oy, cybersecurity researcher with extensive academic and professional experience in computer science, network security, and intelligent computing. He received his Ph.D. in Computer Science, with a research focus on mathematics and cybersecurity, from Guangzhou University, China, in 2020. His research interests encompass cybersecurity, cloud computing, Software-Defined Networking (SDN), big data analytics, Intrusion Detection Systems (IDS), the Internet of Things (IoT), Moving Target Defence (MTD), machine learning, and optimization algorithms. Throughout his academic and professional career, he has collaborated with international researchers and contributed to numerous peer-reviewed journal articles and conference papers addressing emerging challenges in cybersecurity, intelligent networks, cloud infrastructures, and data-driven computing systems. Dr. Javadpour has served as a reviewer for several internationally recognized journals, including IEEE Transactions on Cloud Computing, IEEE Transactions on Network Science and Engineering, ACM Transactions on Internet Technology, and various journals published by Springer and Elsevier. He has also actively contributed to the international research community as a Technical Program Committee (TPC) member for several academic conferences. His international research experience includes active participation in European-funded research projects and consortia, including INSPIRE-5Gplus, RIGOUROUS, 6G-SANDBOX,CHARITY and 6G-PATH. Through these collaborations, he has contributed to research published in leading journals and conferences, including IEEE Transactions on Industrial Informatics, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Network and Service Management, ACM Transactions on Sensor Networks, and IEEE GLOBECOM. In addition to his research activities, Dr. Javadpour has experience mentoring and supervising master's and doctoral students. His academic leadership, international collaborations, participation in funded research projects, and experience in conducting independent research have strengthened his ability to lead multidisciplinary research teams and develop innovative solutions for complex cybersecurity challenges. His current research focuses on secure, intelligent, and resilient computing systems, with particular emphasis on next-generation networks, cloud-edge infrastructures, AI-driven cybersecurity, adaptive defence mechanisms, and trustworthy intelligent systems. Dr. Javadpour has also been recognized among the World's **Top 2% Scientists** in the Stanford University ranking, reflecting the scholarly impact, international visibility, and sustained influence of his research contributions.

FOROUGH JA'FARI is a Senior Researcher in cybersecurity and computer science. She received the B.Sc. degree from Sharif University of Technology and the M.Sc. degree in Computer Network Engineering from Yazd University, Iran. She is a visiting scholar researcher at Guangzhou University, China. Her research interests include cloud computing, software-defined networking (SDN), cyber deception, intrusion detection systems (IDS), the Internet of Things (IoT), moving target defense (MTD), and machine learning methods for proactive and adaptive security. She has contributed to research and peer-review activities across journals and conferences in these areas. She currently serves as a Guest Editor of the Cluster Computing (CLUS) journal and acts as a reviewer for several journals and conferences, with a focus on practical, deployable security mechanisms and data-driven defenses for modern networked systems.

CHAFIKA BENZAÏD is a Senior Research Fellow at the University of Oulu, Finland. From November 2018 to December 2021, she was a Senior Researcher at Aalto University, Finland. Prior to that, she served as an Associate Professor at the University of Sciences and Technology Houari Boumediene (USTHB), Algeria. She received the Engineer, Magister, and Doctorat ès Sciences degrees from USTHB. Her research interests include 5G/6G networks, SDN, network security, AI security, and AI/ML techniques for zero-touch security management and automation. Her work addresses the security and reliability challenges of softwarized and programmable networks, with emphasis on scalable orchestration and policy enforcement. She is an ACM Professional Member.

TARIK TALEB Prof. Tarik Taleb received the B.E. degree with distinction in Information Engineering and the M.Sc. and Ph.D. degrees in Information Sciences from Tohoku University, Sendai, Japan, in 2001, 2003, and 2005, respectively. He is currently a Full Professor at Ruhr University Bochum (RUB), Germany, where he leads research activities on next-generation mobile and distributed systems. Prior to joining RUB, he was a Professor at the University of Oulu, Finland (2018–2023), and an Associate Professor at Aalto University, Finland (2014–2021). Earlier in his career, he served as a Senior Researcher and 3GPP Standards Expert at NEC Europe Ltd., Heidelberg, Germany, where he contributed to the evolution of mobile network architectures and standardization activities. Before joining NEC, he was an Assistant Professor at the Graduate School of Information Sciences, Tohoku University, Japan, working in a research laboratory fully funded by KDDI. He also held a Research Fellowship at the Intelligent Cosmos Research Institute, Japan, from 2005 to 2006. Prof. Taleb is widely recognized for his pioneering contributions to mobile network softwarization, network slicing, cloud-edge continuum management, and autonomous networking. His current research interests include autonomous network and service management, edge-cloud continuum systems, network softwarization and slicing, software-defined security, and AI-native communication networks.