Article

Deep Generative Models for 3D Content Creation: A Comprehensive Survey of Architectures, Challenges, and Emerging Trends

Kaiqi Chen * and Libby Ramsey

Posted Date: 30 October 2024

doi: 10.20944/preprints202410.2397.v1

Keywords: Computer Vision; 3D Generation

 doi:10.20944/preprints202410.2397.v1

Article

Deep Generative Models for 3D Content Creation: A Comprehensive Survey of Architectures, Challenges, and Emerging Trends

Kaiqi Chen * and Libby Ramsey

Independent Researcher, USA

* Correspondence: kaiqichen115@gmail.com

Abstract: The field of 3D model generation has become essential across various industries, including gaming, virtual and augmented reality (VR/AR), architecture, and medical imaging. Traditionally reliant on manual efforts, 3D content creation is now being transformed by deep generative models, enabling more efficient, scalable, and dynamic generation of complex shapes and environments. This survey provides a comprehensive review of key backbone architectures used for 3D generation, including autoencoders, variational autoencoders (VAEs), generative adversarial networks (GANs), autoregressive models, diffusion models, normalizing flows, attention-based models, CLIP-guided models, and procedural generation techniques. We explore each model's role in 3D generation, highlighting their strengths—such as the precision of VAEs, the realism of GANs, the stability of diffusion models, and the scalability of procedural methods—alongside their limitations, such as training instability, high computational costs, and the difficulty in handling multi-modal data. Additionally, we discuss the increasing relevance of attention-enhanced models and the integration of text-based CLIP supervision for improved semantic alignment in 3D outputs. The survey concludes with an analysis of open challenges, including balancing efficiency with expressiveness, managing training complexity, and addressing dataset limitations [1]. It also identifies future research directions, such as few-shot learning, hybrid architectures, and neural-symbolic approaches, which promise to advance the field by improving the generalization and versatility of 3D generation models. This paper aims to guide researchers and practitioners in navigating the evolving landscape of 3D generative methods and inspire new innovations in the creation of realistic, high-quality 3D content.

Keywords: computer vision; 3D Generation

1. Introduction

The field of 3D model generation has become a pivotal area of research due to its wide range of applications, including video games, virtual and augmented reality, digital content creation, autonomous systems, architecture, and medical imaging. Traditionally, 3D model generation required manual work by artists or engineers, but with the advancement of machine learning, deep generative models are transforming how 3D objects are created, modified, and represented. This shift has unlocked new possibilities for creating high-quality 3D content efficiently and dynamically.

Several recent works have addressed specific challenges in 3D generation through innovative model architectures. For example, the Wasserstein Generative Adversarial Network (WG-CNN) was applied to bearing fault diagnosis tasks, illustrating how generative adversarial networks (GANs) and convolutional neural networks (CNNs) can be combined for practical engineering applications [2]. Furthermore, CNN-LSTM models with attention mechanisms have been used for real-time tasks like partial discharge detection [3] and robotic ground classification [4], demonstrating the adaptability of these models across various 3D tasks, and night-time breathing disorder detection [5]. This adaptability showcases the wide utility of these models across diverse 3D tasks, including medical applications such as pneumonia classification from X-ray images [6,7]. In scientific applications, risk-averse state estimation models have been developed for tasks such as GNSS precise point positioning. These models help to accommodate outliers and ensure robust estimations [8], which is crucial in high-precision applications like satellite navigation and autonomous systems.

 doi:10.20944/preprints202410.2397.v1

The use of attention mechanisms and deep learning architectures has also enhanced cybersecurity and defense applications. For instance, attention-enhanced methods have been applied to bolster AI's role in data security and risk control models, particularly in addressing complex 3D data generation tasks [9–11]. Similarly, there have been developments in AI-driven solutions for large-scale 3D generation in the context of risk control and defense systems [10]. In more experimental applications, research has explored AI techniques such as speculative sampling in reasoning tasks [12,13], and improved ReAct frameworks [14], demonstrating that even reasoning processes can be optimized using generative models. Additionally, developments in game strategy simulation [15] and AI-generated text detection [16,17] illustrate the breadth of 3D model applications across various domains.

Unlike 2D image generation, 3D model generation is inherently more complex, requiring models to capture not just spatial relationships but also volumetric consistency, surface smoothness, and multi-view coherence. As a result, several unique challenges arise, including the representation of 3D data (e.g., voxel grids, point clouds, meshes, implicit neural representations) and balancing the trade-off between quality, memory efficiency, and computational demands.

Creating 3D models involves addressing several key challenges. First, 3D data requires more memory and processing power than 2D data, making it harder to model and compute. In addition, unlike 2D images, 3D models can be represented through multiple formats, such as voxels, point clouds, or meshes. This diversity complicates the development of unified generative models. Furthermore, models must ensure that generated objects are visually coherent from multiple viewpoints and maintain correct geometric structure [18,19]. High-quality 3D datasets are limited, and generating labeled datasets requires significant manual effort.

Given the rapid evolution in this domain, this paper aims to consolidate the latest developments and provide a thorough understanding of the backbone architectures used for 3D generation. The primary contributions of this paper are:

- **Comprehensive Review of Backbone Architectures:**

  We present a detailed survey of the most influential backbone models, including autoencoders, variational autoencoders (VAEs), generative adversarial networks (GANs), attention-based models, autoregressive models, diffusion models, CLIP-guided architectures, normalizing flows, and procedural models. Each architecture is discussed with respect to its role in 3D generation, its key strengths, and its limitations. Similarly, attention mechanisms in assistive navigation systems such as Assister [20] and generalized visual odometry systems like XVO [21,22] highlight their utility in real-time guidance tasks.
- **In-depth Analysis of Model Strengths and Weaknesses**: For each architecture, we identify the specific advantages that make it suitable for particular 3D generation tasks and the shortcomings that limit its broader applicability. This analysis will guide researchers and practitioners in choosing the appropriate models for their use cases.
- **Exploration of New and Emerging Trends:** We highlight recent trends such as attention-enhanced GANs and diffusion models, as well as the growing role of multi-modal models like CLIP [23,24]. We also explore the increasing use of procedural methods as a complementary approach for scalable 3D generation.
- **Discussion of Open Challenges and Research Directions:** We discuss key open problems and challenges faced by the research community, such as balancing model efficiency and expressiveness, developing multimodal and cross-domain generation techniques, and addressing ethical concerns [25]. Additionally, we outline promising directions for future work, including few-shot and zero-shot 3D generation and new representations combining neural fields with symbolic rules.

By providing a detailed taxonomy and breakdown of the various backbone models, this survey offers both novice and experienced researchers a valuable resource to understand the landscape of 3D generation. We aim to support the community in navigating this rapidly evolving field and inspire new innovations that will further advance the creation of realistic, high-quality 3D models.

 doi:10.20944/preprints202410.2397.v1

Table 1. Summary of Key Models and Their Characteristics in 3D Generation.

Model Type	Key Characteristics	Strengths	Challenges	Applications
Voxel Grids	3D grid representation, spatially structured	Compatible with CNNs, simple structure	High memory usage, blocky representations	Object classification, segmentation, volumetric reconstruction, MRI/CT scans
Point Clouds	Sparse set of 3D points, flexible format	Memory-efficient, retains surface details	Lacks surface connectivity, requires specialized processing	3D scanning, LiDAR-based perception, AR/VR, autonomous driving
Polygonal Meshes	Vertices, edges, faces representation	Accurate surface modeling, compatible with graphics tools	High polygon count can be resource-intensive	Character modeling, architectural rendering, CAD, VR, gaming
Implicit Representations	Function-based 3D object representation	Continuous surfaces, scalable detail	Slow inference, complex transformations	Real-time guidance, shape interpolation, high-resolution reconstruction
Procedural Generation	Rules/algorithms-based generation	Scalable, minimal human intervention	Limited control, complex to fine-tune	Gaming, architectural design, VR environments, crowd generation
Autoencoders (AEs)	Encoder-decoder structure, compresses and reconstructs	Effective at shape morphing, dimensionality reduction	Struggles with fine details, limited expressiveness	Dimensionality reduction, anomaly detection, shape reconstruction
Variational Autoencoders (VAEs)	Probabilistic extension of AEs, Gaussian distribution learning	Smooth interpolation, diverse outputs	Blurry outputs, lacks photorealism	Morphing, exploratory design, few-shot learning, medical imaging
Generative Adversarial Networks (GANs)	Adversarial training (generator and discriminator)	High realism, flexible structure	Training instability, mode collapse	Text-to-3D generation, gaming assets, character design, virtual environments
Attention-Based Models	Utilizes self-attention to capture dependencies	High detail, complex dependency modeling	Memory-intensive, complex training	Real-time navigation, shape detail capture, complex object generation
Autoregressive Models	Sequential generation, conditional dependencies	Precision control, ideal for incremental generation	Slow inference, prone to error propagation	Shape completion, intricate 3D structures, real-time applications
Diffusion Models	Iterative denoising, gradually refines outputs	Stable training, high fidelity	High computational cost, slow generation	High-quality point cloud generation, 3D reconstruction, medical imaging
CLIP-Guided Models	Multimodal (text-image) association, embedding alignment	Semantic alignment, creative control	Limited text diversity, computational overhead	Text-to-3D generation, creative design, VR asset creation
Normalizing Flows	Sequence of invertible transformations, exact probability modeling	High precision, continuous latent space	Invertibility constraint, architectural complexity	Density estimation, point cloud generation, interpolation, scientific simulations

2. Background Info & Taxonomy Explanation

Different data representations offer unique ways to encode 3D models. Each format has its trade-offs, impacting memory usage, resolution, expressiveness, and computational efficiency.

2.1. Voxel Grids

Voxel grids represent 3D objects or scenes by dividing the space into a regular grid of small cubic units called voxels, analogous to pixels in 2D images. Each voxel stores information about whether a specific portion of the 3D space is occupied or empty. In some cases, voxels may contain additional attributes such as density values, color, or material properties. This representation is straightforward and widely used in early 3D applications because it extends naturally from 2D image processing techniques, making it compatible with convolutional neural networks (CNNs) and other grid-based

 doi:10.20944/preprints202410.2397.v1

methods [26]. This representation is straightforward and widely used in early 3D applications because it extends naturally from 2D image processing techniques, making it compatible with convolutional neural networks (CNNs) and other grid-based methods [27].

The key strength of voxel grids lies in their simplicity and compatibility with deep learning models [28]. Operations like 3D convolution can be easily applied to these grids, allowing networks to extract spatial features and generate volumetric structures. Voxel grids are particularly suitable for tasks such as object classification, segmentation, and volumetric reconstruction [29], especially when working with medical imaging datasets like MRI or CT scans. Furthermore, voxel-based models align well with physical simulations, as the grid structure provides an easy way to represent spatial relationships and interactions in 3D space [30].

Despite their advantages, voxel grids have significant limitations. The memory requirements grow cubically with resolution, meaning that even moderate-resolution grids can become prohibitively large. For example, a 128x128x128 voxel grid contains over two million voxels, making storage and computation inefficient [31,32]. As a result, voxel-based models often need to operate at lower resolutions, which can compromise the level of detail in the generated models. To address this, researchers have explored sparse voxel representations and hierarchical grids to reduce memory overhead, but these techniques introduce additional complexity.

Voxel grids are also limited in their ability to capture fine surface details. Unlike meshes, which can explicitly model smooth surfaces and geometric intricacies, voxel grids only approximate the shape of objects by filling in volume. As a result, small features may be lost, and the generated objects can appear blocky or coarse. While post-processing techniques such as marching cubes can extract smoother surfaces from voxel data, these steps add extra computational burden to the generation pipeline.

2.2. Point Clouds

Point clouds represent 3D objects as a set of discrete points distributed in space, with each point having specific 3D coordinates (x, y, z) and potentially other attributes like color, normal vectors, or intensity. Unlike voxel grids, which discretize space into a fixed-size grid, point clouds offer a more compact and flexible way of representing shapes and surfaces. This data structure is widely used in applications such as 3D scanning, LiDAR-based perception for autonomous vehicles [33–35], augmented reality (AR), and virtual reality (VR), where capturing objects and environments with sparse or incomplete data is common.

One of the main advantages of point clouds over voxel grids is their efficiency in memory and computation. While voxel grids require cubic memory growth with resolution (e.g., a 128x128x128 voxel grid stores over 2 million voxels), point clouds focus only on the relevant regions of an object by storing a sparse set of points that define its surface. This efficiency allows point cloud models to scale better to higher resolutions without consuming excessive memory. As a result, point clouds are preferred when working with large datasets or real-time systems such as LiDAR-based object detection, where fast and efficient processing is essential.

Another reason point clouds are superior to voxel grids is their ability to capture fine surface details. Voxel grids approximate objects by filling space with uniform cubes, which can lead to blocky and coarse representations, especially at lower resolutions. In contrast, point clouds directly encode surface points, making them more adept at preserving the intricate geometry of objects. This property is particularly valuable in applications such as architectural design, robotics, and 3D scanning, where accuracy and detail are paramount [36,37].

Point clouds are also more flexible for representing incomplete or non-uniform data. In real-world scenarios, 3D data collection often results in incomplete or sparse datasets (e.g., a partial scan of an object from a specific angle). Voxel grids, which require a complete 3D space to be filled, struggle with incomplete data, leading to noisy or ambiguous outputs. On the other hand, point clouds can naturally handle partial observations and irregular sampling, making them ideal for tasks where data is gathered

 doi:10.20944/preprints202410.2397.v1

from multiple views or sensors. Additionally, point clouds are invariant to scale and rotation, which simplifies downstream processing and reduces the need for data normalization [38].

However, point clouds also have some limitations. One major challenge is that point clouds lack explicit connectivity information, meaning there is no inherent knowledge of how the points form surfaces or shapes. Unlike meshes, which store vertices and edges to define surfaces, point clouds require additional processing (e.g., surface reconstruction algorithms like Poisson surface reconstruction) to generate smooth surfaces from the point data [39,40]. This lack of structure can also make it harder for neural networks to process point clouds efficiently, as standard convolutional operations used for voxel grids or images do not apply directly. To address this, researchers have developed specialized architectures, such as PointNet and PointNet++, which are capable of learning from unordered point cloud data. Models like LMSD-YOLO [41] have been effective in multi-scale detection, demonstrating their utility in real-time object detection tasks [42]. Mapping new realities in ground truth creation also showcases the potential for transforming image-to-image translations in 3D content generation, as observed with models like pix2pix [43].

2.3. Polygonal Meshes

Polygonal meshes are one of the most popular and widely used representations for 3D models, especially in industries like computer graphics, animation, and CAD (Computer-Aided Design) . A polygonal mesh consists of vertices, edges, and faces, where vertices represent points in 3D space, edges connect these vertices, and faces (typically triangular or quadrilateral) form the surface of the 3D object. This representation is particularly well-suited for applications requiring detailed surface modeling, such as 3D printing, video games, and visual effects in movies.

One of the biggest strengths of polygonal meshes is their ability to accurately represent surfaces and fine details. Since the faces of a mesh follow the contours of an object, it can precisely capture intricate geometries and smooth surfaces. This makes meshes ideal for high-fidelity applications, such as character modeling in video games, architectural rendering, and complex industrial designs. Compared to voxel grids, which discretize the space and may result in blocky approximations, meshes can achieve a high level of geometric precision without excessive memory consumption [34].

Meshes also allow easy manipulation of surfaces. Through operations like subdivision, smoothing, and deformation, artists and engineers can precisely adjust mesh geometry to achieve the desired shape and detail. Furthermore, polygonal meshes are well-supported by industry-standard tools and software, including Blender, Maya, and AutoCAD, making them the go-to format for professionals in the 3D design space. This extensive software ecosystem facilitates the integration of meshes into production pipelines for gaming, animation, and virtual reality [33,44].

Another important feature of polygonal meshes is their efficiency in rendering and visualization [35,45]. Modern graphics processing units (GPUs) are optimized to render polygonal surfaces efficiently, particularly triangular meshes. As a result, polygonal meshes are the backbone of most 3D rendering engines, including those used in games (e.g., Unreal Engine) and virtual worlds. The combination of high expressiveness and real-time rendering capabilities makes them an essential part of both real-time and offline rendering workflows [46,47].

Despite their advantages, polygonal meshes also present some challenges. One of the primary issues is that meshes can become complex and unwieldy as the number of polygons increases. High-resolution meshes with millions of polygons may require significant computational resources for storage, manipulation, and rendering. This can become a bottleneck in applications that require real-time interaction, such as VR or AR systems. To address this, novel approaches such as location-refined feature pyramid networks (LR-FPN) have been proposed to enhance object detection [48]. Additionally, texture extraction methods, like those applied to sedimentary structures classification [49,50], demonstrate the importance of feature extraction in 3D representation tasks.

 doi:10.20944/preprints202410.2397.v1

3. Implicit Representations

Implicit representations, often referred to as neural fields, offer a powerful and flexible way of modeling 3D objects by representing them through continuous mathematical functions rather than discrete elements such as voxels, points, or polygons. The essence of implicit representations lies in their ability to encode complex surfaces by learning a function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ that maps 3D coordinates to a scalar value, such as signed distance values or occupancy probabilities. This function-based approach has gained significant traction with the advent of deep neural networks [51], which are now used to approximate these implicit functions, enabling smooth and high-quality surface generation.

In implicit models, instead of explicitly storing the geometry of an object, the surface is defined as the level set of a function. One popular type of implicit representation is the Signed Distance Function (SDF), where the function value at a given point indicates the shortest distance to the surface, with the sign determining whether the point is inside or outside the object. Another common type is the occupancy field, which predicts whether a given point in 3D space lies inside the object (occupancy value close to 1) or outside (occupancy value close to 0).

These neural fields are often represented using fully connected neural networks, typically referred to as Multi-Layer Perceptrons (MLPs). The MLP learns to approximate the underlying continuous function by training on samples of 3D points and their corresponding values (e.g., distances or occupancy probabilities). Once trained, surfaces can be extracted using algorithms such as Marching Cubes, which identifies the zero-crossing points of the learned function and reconstructs the mesh.

4. Procedural Representations

Procedural representations generate 3D models using a set of rules, algorithms, and mathematical functions rather than direct data-driven approaches like deep learning. In procedural generation, objects or scenes are created dynamically based on predefined parameters, often introducing an element of randomness or controlled variation. This method has long been used in gaming, architecture, urban modeling, virtual environments, and simulations, where creating vast amounts of complex content manually is infeasible. Procedural techniques can generate anything from natural landscapes, cities, and trees to intricate textures and entire virtual worlds, making them essential for content generation at scale.

At the core of procedural generation lies a set of rules or algorithms that guide how elements are combined or structured to produce 3D objects. These rules can involve randomization, grammar-based methods, fractals, noise functions (e.g., Perlin noise), or parametric modeling, depending on the nature of the task [52,53]. In many cases, seed values are used to initialize the algorithm, ensuring that each generation is unique but still follows the general principles defined by the rules [54]. Additionally, methods such as intelligent vehicle classification using deep learning and multisensor fusion [55,56] demonstrate how hybrid architectures can be applied to specific 3D tasks like autonomous driving and classification systems [12,57].

A well-known example is the Lindenmayer system (L-system), a type of procedural grammar used to generate organic structures like plants or branching patterns. In an L-system, a set of recursive rules defines how the structure grows step-by-step, mimicking natural processes [58]. Similarly, Perlin noise and other noise-based algorithms are used to generate terrain or textures, giving rise to realistic landscapes and surfaces. In parametric design, objects are defined using mathematical equations or parameters, which makes it easy to create multiple variations of a model by tweaking the input parameters. Research into cancer gene data classification has also illustrated procedural methods' role in enhancing the robustness of classification frameworks [59–61].

5. 3D Generation Models

The development of generative models for 3D data has resulted in a variety of approaches, each with unique characteristics. We can classify these models into several core categories, based on their

 doi:10.20944/preprints202410.2397.v1

underlying architectures and methods. Below is a taxonomy of the primary classes of generative models used for 3D generation [62,63].

5.1. Autoencoders and Variational Autoencoders (VAEs)

Autoencoders are neural networks that aim to learn a compact latent representation of input data and use this representation to reconstruct the original input. The architecture of an autoencoder consists of two main components: the encoder and the decoder. The encoder compresses the high-dimensional input data into a smaller latent space, effectively capturing the most essential features of the data. The decoder then takes the latent representation and tries to reconstruct the original input as accurately as possible. Autoencoders are widely used in tasks such as dimensionality reduction, denoising, anomaly detection [64], and 3D shape reconstruction, where learning meaningful patterns from data is essential [65–67].

Variational Autoencoders (VAEs) extend the basic concept of autoencoders by incorporating a probabilistic framework. In a standard autoencoder, the encoder maps each input to a fixed latent vector. However, in a VAE, the encoder maps the input to mean and variance parameters of a Gaussian distribution in the latent space, allowing the model to learn distributions rather than fixed embeddings. During the generation process, new samples can be produced by sampling from these latent distributions, which makes VAEs more suitable for tasks such as generative modeling, interpolation, and smooth shape transitions. The probabilistic nature of VAEs ensures that similar inputs are mapped to nearby points in the latent space, enabling the generation of smooth and coherent outputs when interpolating between different 3D shapes.

Autoencoders are particularly effective in 3D reconstruction tasks, where objects represented as voxel grids, point clouds, or meshes are compressed into a latent space and reconstructed with minimal loss of information. However, the limitation of standard autoencoders lies in their tendency to focus on global structures rather than fine-grained details. This often results in reconstructed 3D objects that capture the overall shape but lack sharp edges or intricate surface features. The bottleneck in the latent space—designed to compress information—can cause loss of subtle features, making it difficult for autoencoders to handle tasks that require high levels of detail, such as intricate character models or complex architectural elements.

Variational autoencoders address some of these limitations by regularizing the latent space through a process called KL-divergence loss, which ensures that the learned latent distribution resembles a standard Gaussian distribution. This regularization allows VAEs to produce more coherent and diverse samples, even for objects with complex variations. However, VAEs still struggle with blurry outputs and may not capture all fine details when compared to more advanced generative models like GANs. This trade-off between stability (VAEs train more easily than GANs) and expressiveness (GANs tend to generate sharper results) makes VAEs a good choice for exploratory tasks such as interpolation and shape morphing, but less ideal for tasks that require photorealistic outputs.

One of the most promising aspects of VAEs is their ability to smoothly interpolate between different 3D shapes. For instance, given two latent vectors representing distinct objects (e.g., a chair and a table), the VAE can generate intermediate shapes that transition smoothly between the two. This property makes VAEs highly useful in creative design tasks and morphing applications, where designers need to explore a continuum of possibilities between different objects. Additionally, VAEs are increasingly being integrated into hybrid architectures, combining them with other models like GANs or transformers to balance stability and output quality.

Despite their advantages, VAEs face challenges in high-resolution 3D generation due to the inherent trade-off between stability and output sharpness. Researchers are actively exploring improvements to VAE architectures, such as hierarchical VAEs and more expressive priors, to address these limitations. Additionally, there is growing interest in applying VAEs to conditional generation tasks, where the generation of 3D objects is guided by input constraints, such as text descriptions or partial 3D scans.

 doi:10.20944/preprints202410.2397.v1

This line of research opens up exciting possibilities for few-shot learning and guided shape generation, which are critical for applications in gaming, virtual reality, and robotics [68].

5.2. Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) consist of two key neural networks that compete during the training process. The generator network creates synthetic data designed to resemble real data, while the discriminator network attempts to distinguish between the real data and the fake data generated by the generator [69,70]. This adversarial process pushes both networks to improve: the generator learns to create more realistic outputs, and the discriminator becomes better at spotting subtle flaws. This framework has achieved significant success across domains such as image synthesis, music generation, and text-based data creation, and is now being extended into 3D data generation [71**?**].

Adaptation to 3D Data Generation In recent years, GANs have been adapted to generate 3D shapes, contributing to fields like virtual reality (VR), gaming, and design. Unlike 2D GANs, which operate on image pixels, 3D GANs work with 3D data structures, such as voxel grids, meshes, point clouds, and implicit surfaces like neural radiance fields (NeRF) [58,72,73]. The two main architectures applied to 3D generation include voxel-based GANs and mesh-based GANs, each with unique benefits and challenges [74].

Voxel-based GANs generate 3D models by organizing objects as 3D grids of volumetric pixels (voxels). An example is the MSG-Voxel-GAN, which improves training stability by using multi-scale gradients and can generate detailed, high-resolution 3D shapes. However, voxel-based GANs are computationally intensive due to the high memory requirements of 3D convolutions, limiting their scalability to complex scenes [75]

Mesh-based GANs produce 3D objects using connected vertices and triangular faces, making them highly compatible with computer graphics tools. A notable example is NVIDIA's XDGAN, which parameterizes 3D objects into 2D geometry images. This method leverages 2D GAN architectures like StyleGAN to generate 3D content efficiently by transforming geometry images back into textured meshes. This hybrid approach ensures high visual fidelity while remaining computationally efficient, addressing some limitations of voxel-based GANs.

The primary use case for 3D GANs is generating high-quality and diverse 3D shapes. These models are essential in several industries, such as gaming and virtual reality (VR). Developers use 3D GANs to generate assets for immersive experiences and reduce the manual effort needed to create characters, environments, and props. In Computer-Aided Design (CAD), GANs assist in automating product design by generating innovative shapes that adhere to predefined specifications. 3D generation techniques have also been found to be extensively useful in medical imaging, where detailed reconstructions are critical. For example, deep learning-based models have been used for multimodal MRI reconstruction and synthesis [76], as well as for efficient parallel MRI reconstruction [77]. Optimization-based deep learning methods have also been developed for magnetic resonance imaging (MRI) [78]. These methods are particularly important in scenarios requiring high precision, such as brain imaging and tumor detection [79].

Additionally, some 3D GAN models, such as EG3D and Mimic3D, integrate text-based models to enable text-to-3D generation, allowing designers to create objects or scenes from natural language descriptions, which can streamline creative processes. In more complex scenarios, hybrid models like Bayesian-Optimized Attentive Neural Networks (BOANN) are being explored for classification tasks [80], and MPGAAN, a heterogeneous information network classification model, has been proposed as an efficient solution for network-based classification problems [81].

5.3. Attention-Based Models in 3D Data Generation

Attention mechanisms have become essential for 3D data generation, improving the capture of intricate details by selectively focusing on critical parts of an object. These mechanisms are integrated

 doi:10.20944/preprints202410.2397.v1

into different backbone architectures, including transformer-based models [82], GANs, and hybrid implicit models.

5.3.1. Transformer-Based Models

Transformers, which process input elements as tokens, excel at modeling spatial relationships within 3D data. Models such as 3D ShapeFormer and Point Transformer utilize self-attention to capture dependencies between points or regions in a 3D shape. This enables the generation of highly detailed objects, especially for point cloud data [83]. However, transformers have high memory and computational demands, as self-attention scales quadratically with input size, limiting their efficiency for large datasets.

5.3.2. GAN-Based Architectures with Attention

GANs enhanced with attention mechanisms, such as EG3D and XDGAN, are designed to generate detailed surfaces by focusing on specific areas of the 3D structure. EG3D uses attention to ensure consistency across multiple views, while XDGAN leverages 2D-to-3D imitation through geometry images to align surface textures. These models excel at generating fine-grained details but introduce additional parameters, increasing training complexity. Despite their capabilities, they remain vulnerable to mode collapse, producing limited variations if not properly balanced.

5.3.3. Hybrid Implicit Models with Attention

Implicit models like NeRF and DeepSDF encode objects as continuous fields and employ attention to prioritize high-detail areas, such as edges. These architectures achieve sharp feature preservation, but their reliance on dense computations makes inference slow. Additionally, converting implicit representations to mesh or voxel formats introduces overhead, which limits their immediate usability in interactive 3D applications.

5.4. Autoregressive Models

Autoregressive models generate data sequentially, one element at a time, by conditioning each output step on the previously generated elements. This approach has proven effective in tasks such as text generation and 2D image synthesis, and it is now being extended to 3D data generation. In the context of 3D, these models predict objects or structures part-by-part or slice-by-slice, maintaining complex dependencies throughout the sequence.

These models are particularly useful in 3D reconstruction, generating missing parts of objects based on observed data, and in shape generation, where intricate structures like architectural elements are produced incrementally. Some models employ attention mechanisms, similar to transformers, to capture dependencies across points or voxels, ensuring coherence in complex shapes.

The primary strength of autoregressive models lies in their fine-grained control over the generation process, making them ideal for tasks requiring precision, such as detailed shape completion. However, they suffer from slow inference due to their step-by-step nature and are prone to cumulative errors, where mistakes early in the sequence propagate through the entire output.

5.5. Diffusion-Based Models

Diffusion-based models generate data through an iterative process, gradually refining noisy inputs into coherent structures. These models start with a noisy or random distribution and incrementally "denoise" it, allowing for the creation of complex and detailed 3D shapes. Originally successful in 2D image synthesis, diffusion models have recently been extended to 3D tasks, leveraging their ability to capture intricate patterns.

Diffusion models are especially useful for generating high-quality point clouds, voxel grids, and meshes, with applications in areas like shape generation and 3D reconstruction. In text-to-3D pipelines, diffusion models provide consistent, high-fidelity outputs by iteratively refining the structure until

 doi:10.20944/preprints202410.2397.v1

it matches the target prompt or design specifications. They are also used in medical imaging, where precise 3D reconstructions are essential.

The main strength of diffusion models lies in their stability and ability to capture fine details. By iteratively refining outputs, these models avoid some common pitfalls seen in GANs, such as mode collapse. They also produce more diverse outputs, making them well-suited for generating multiple variations of a 3D object.

However, diffusion models are computationally intensive due to the large number of iterative steps required for denoising, resulting in slower inference. They also demand substantial memory and processing power, limiting their scalability for real-time applications or high-resolution 3D environments.

5.6. CLIP (Contrastive Language-Image Pretraining)

CLIP (Contrastive Language-Image Pretraining) is a multimodal model developed by OpenAI that learns to associate text with images [84]. It leverages a large-scale dataset of text-image pairs to align their respective embeddings in a shared latent space. In 3D generation, CLIP is used to guide models, such as GANs or diffusion models, by providing semantic supervision through text prompts [85]. This allows for the creation of 3D objects based on textual input, enabling text-to-3D generation with better alignment between shape, style, and semantics.

CLIP enhances 3D generation models by aligning generated objects with natural language descriptions. This is crucial in text-driven 3D asset creation, where users provide textual descriptions (e.g., "a futuristic car") and the model generates a 3D representation that matches the description.

5.6.1. Text-Guided Diffusion Models

Some recent 3D pipelines employ CLIP to refine 3D structures through iterative feedback. For example, a diffusion model can generate rough 3D shapes that are evaluated and adjusted based on the similarity between the generated object and the given text description in CLIP's embedding space.

5.6.2. CLIP-Guided GANs

GAN-based models also integrate CLIP to improve consistency between generated 3D shapes and prompts. By measuring alignment in the latent space, CLIP helps enforce semantic accuracy, ensuring that generated outputs closely match the intended concept [61].

One of the primary strengths of CLIP is its ability to ensure semantic alignment between textual descriptions and generated 3D objects. By embedding both text and visual data into a shared latent space, CLIP allows 3D models to produce outputs that better capture the intent behind natural language prompts. This capability significantly improves creative control, enabling designers to generate specific 3D shapes that align with user expectations. Additionally, CLIP's flexibility across different domains makes it effective for a variety of applications, including game design, virtual reality (VR) asset creation, and rapid prototyping. Its capacity to interpret and process a wide variety of concepts helps models generate not only realistic objects but also more abstract or artistic forms.

However, CLIP is not without limitations. Since its performance depends heavily on the quality and diversity of the text-image pairs used during pretraining, it may struggle with niche or complex prompts, leading to inconsistent results. In scenarios where the intended meaning of a prompt is ambiguous or highly specialized, the generated output might fail to meet expectations. Another significant limitation is the computational overhead that comes with using CLIP in conjunction with 3D models [86]. Many 3D generation pipelines require iterative feedback loops to refine outputs, and incorporating CLIP into this process can make it computationally intensive, thus limiting its use in real-time applications like interactive design environments or dynamic VR settings. As a result, optimizing CLIP for efficiency while maintaining alignment quality remains an ongoing challenge in research [87].

 doi:10.20944/preprints202410.2397.v1

5.7. Normalizing Flows

Normalizing flows are a class of generative models that transform a simple probability distribution (e.g., a Gaussian) into a more complex target distribution through a sequence of invertible transformations [88,89]. These transformations allow flows to generate complex outputs while maintaining exact likelihood estimation, making them suitable for tasks where precise probability modeling is essential, such as 3D data generation [90,91].

In 3D generation, normalizing flows excel at tasks like density estimation and point cloud generation. They model the data distribution directly, enabling the generation of highly detailed and structured objects [92]. Some flow-based models generate 3D point clouds by sequentially transforming points from a standard Gaussian distribution into realistic 3D shapes, maintaining continuous control over both global and local features. Additionally, they are used for 3D shape interpolation, smoothly transforming one 3D object into another, useful in applications like morphing animations or shape blending in virtual environments.

A major strength of normalizing flows is their ability to provide exact likelihood estimation, which allows them to model complex 3D data distributions accurately. This feature makes them particularly effective for applications requiring high precision, such as scientific simulations or medical imaging [93]. Furthermore, their invertibility ensures that every transformation is reversible, making it possible to both generate realistic outputs and analyze the underlying distribution with precision. Additionally, flows offer continuous latent spaces, which are beneficial for tasks like interpolation and controlled shape generation [94].

Despite their strengths, normalizing flows come with some limitations. They can be computationally intensive, as each transformation step in the sequence must remain invertible, which constrains model architecture. This makes it challenging to scale flows for very high-dimensional data, such as voxel grids or high-resolution 3D models. Moreover, designing effective transformations for complex 3D objects requires significant expertise, limiting the ease of use compared to other models like GANs or diffusion models. Another challenge is the potential sensitivity to initialization and parameterization, which can impact training stability and performance.

5.8. Procedural Generation in 3D Models

Procedural generation refers to the creation of 3D content using predefined rules, algorithms, and parameters rather than manual design. This method allows for the efficient creation of complex structures, such as landscapes, cities, or intricate patterns, with minimal human intervention. Rules guide the generation process by defining relationships between components, ensuring consistency while also enabling variability [95,96].

Rules in procedural generation often describe the structure and behavior of objects at different levels. For example, in a city generation model, rules may govern how buildings are placed along roads, ensuring logical connections between streets and intersections. Similarly, for terrain generation, algorithms like fractals or Perlin noise generate natural-looking landscapes by specifying height distributions and surface features.

A key aspect of procedural rules is their flexibility. Simple parametric controls (e.g., size, shape, material) can lead to extensive variations, allowing designers to generate a wide range of outputs from the same base rules. L-systems (Lindenmayer systems), for instance, are frequently used to generate trees or plants by defining recursive branching rules. These rules enable the creation of complex models that maintain an organic appearance without manual modeling efforts [97,98].

Procedural generation is heavily used in industries like gaming and film. In games, procedural rules create expansive environments (e.g., entire worlds in Minecraft or No Man's Sky) with minimal storage overhead by generating content on the fly. Film studios use procedural techniques to create large crowds, natural scenery, or fantasy cities efficiently. Architecture and urban planning also benefit from procedural rules by automating the generation of building layouts and urban spaces.

 doi:10.20944/preprints202410.2397.v1

6. Conclusions

This survey has provided a comprehensive overview of the key models and methods in the rapidly advancing field of 3D generation. As the demand for high-quality 3D content grows across industries such as gaming, virtual reality (VR), architecture, and medical imaging, deep generative models are transforming the way objects are created, offering more efficient, scalable, and automated solutions [39,99].

The exploration of backbone architectures, from autoencoders and GANs to diffusion models, normalizing flows, and procedural generation, highlights the diversity of approaches available. Each model has unique strengths: GANs excel in generating visually realistic objects, VAEs and autoregressive models provide smooth interpolation and control, and diffusion models offer stability and diversity. Attention mechanisms and multimodal approaches such as CLIP-guided models further enhance generation by aligning outputs with complex text prompts, opening new avenues for creative design.

However, challenges remain, including computational overhead, training stability, and the limited availability of high-quality 3D datasets. Procedural generation offers an alternative by using rule-based algorithms to dynamically generate content, yet it too faces limitations in precision and control. As 3D applications evolve, balancing expressiveness with efficiency will be a key focus, alongside developing methods to integrate cross-domain and multimodal capabilities.

Looking forward, emerging trends such as zero-shot and few-shot learning, hybrid architectures, and neural-symbolic methods promise to push the boundaries of 3D generation [100]. These developments will enable models to generalize better across tasks, generate content with minimal supervision, and seamlessly combine learned representations with symbolic rules. By addressing these challenges, researchers and practitioners can unlock new possibilities in generating photorealistic, detailed, and functionally accurate 3D content, paving the way for future innovations across multiple domains.

References

1. Wenbo Zhu and Tiechuan Hu. Twitter sentiment analysis of covid vaccines. In *2021 5th International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pages 118–122, 2021.
2. Hang Yin, Zhongzhi Li, Jiankai Zuo, Hedan Liu, Kang Yang, and Fei Li. Wasserstein generative adversarial network and convolutional neural network (wg-cnn) for bearing fault diagnosis. *Mathematical Problems in Engineering*, 2020(1):2604191, 2020.
3. Zhongzhi Li, Na Qu, Xiaoxue Li, Jiankai Zuo, and Yanzhen Yin. Partial discharge detection of insulated conductors based on cnn-lstm of attention mechanisms. *Journal of Power Electronics*, 21:1030–1040, 2021.
4. Xiaoxue Li, Jiehong Wu, Zhongzhi Li, Jiankai Zuo, and Pengcheng Wang. Robot ground classification and recognition based on cnn-lstm model. In *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, pages 1110–1113. IEEE, 2021.
5. Bo Dang, Danqing Ma, Shaojie Li, Zongqing Qi, and Elly Zhu. Deep learning-based snore sound analysis for the detection of night-time breathing disorders. *Applied and Computational Engineering*, 76:109–114, 07 2024.
6. Danqing Ma, Yuanfang Yang, Qiyuan Tian, Bo Dang, Zongqing Qi, and Ao Xiang. Comparative analysis of x-ray image classification of pneumonia based on deep learning algorithm algorithm. *Research Gate*, 08 2024.
7. Zhicheng Ding, Panfeng Li, Qikai Yang, and Siyang Li. Enhance image-to-image generation with llava-generated prompts. In *2024 5th International Conference on Information Science, Parallel and Distributed Systems (ISPDS)*, pages 77–81. IEEE, 2024.
8. Wang Hu, Jean-Bernard Uwineza, and Jay A. Farrell. Outlier Accommodation for GNSS Precise Point Positioning using Risk-Averse State Estimation. *arXiv preprint arXiv:2402.01860*, 2024.
9. Yijie Weng and Jianhao Wu. Leveraging artificial intelligence to enhance data security and combat cyber attacks. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, 5(1):392–399, 2024.
10. Yijie Weng and Jianhao Wu. Big data and machine learning in defence. *International Journal of Computer Science and Information Technology*, 16(2), 2024.
11. Yijie Weng, Yongnian Cao, Meng Li, and Xuechun Yang. The application of big data and ai in risk control models: Safeguarding user security. *International Journal of Frontiers in Engineering Technology*, 6(3), 2024.

12. Han Xu, Jingyang Ye, Yutong Li, and Haipeng Chen. Can speculative sampling accelerate react without compromising reasoning quality? In *The Second Tiny Papers Track at ICLR 2024*, 2024.
13. Jinghan Zhang, Xiting Wang, Weijieying Ren, Lu Jiang, Dongjie Wang, and Kunpeng Liu. Ratt: Athought structure for coherent and correct llmreasoning. *arXiv preprint arXiv:2406.02746*, 2024.
14. Shuoqiu Li, Han Xu, and Haipeng Chen. Focused react: Improving react through reiterate and early stop. *arXiv preprint arXiv:2410.10779*, 2024.
15. Ye Zhang, Mengran Zhu, Kailin Gui, Jiayue Yu, Yong Hao, and Haozhan Sun. Development and application of a monte carlo tree search algorithm for simulating da vinci code game strategies. *arXiv preprint arXiv:2403.10720*, 2024.
16. Ye Zhang, Qian Leng, Mengran Zhu, Rui Ding, Yue Wu, Jintong Song, and Yulu Gong. Enhancing text authenticity: A novel hybrid approach for ai-generated text detection. *arXiv preprint arXiv:2406.06558*, 2024.
17. Ye Zhang, Kangtong Mo, Fangzhou Shen, Xuanzhen Xu, Xingyu Zhang, Jiayue Yu, and Chang Yu. Self-adaptive robust motion planning for high dof robot manipulator using deep mpc. *arXiv* 2024, arXiv:2407.12887.
18. Yufeng Zhang, Xue Wang, Longsen Gao, and Zongbao Liu. Manipulator control system based on machine vision. In *International Conference on Applications and Techniques in Cyber Intelligence ATCI 2019: Applications and Techniques in Cyber Intelligence 7*, pages 906–916. Springer, 2020.
19. Longsen Gao, Kevin Aubert, David Saldana, Claus Danielson, and Rafael Fierro. Decentralized adaptive aerospace transportation of unknown loads using a team of robots. *arXiv preprint arXiv:2407.08084*, 2024.
20. Zanming Huang, Zhongkai Shangguan, Jimuyang Zhang, Gilad Bar, Matthew Boyd, and Eshed Ohn-Bar. Assister: Assistive navigation via conditional instruction generation. In *European Conference on Computer Vision*, pages 271–289. Springer, 2022.
21. Lei Lai, Zhongkai Shangguan, Jimuyang Zhang, and Eshed Ohn-Bar. Xvo: Generalized visual odometry via cross-modal self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10094–10105, 2023.
22. Xintao Li and Sibei Liu. Predicting 30-day hospital readmission in medicare patients: Insights from an lstm deep learning model. *medRxiv*, 2024.
23. Xiaojing Fan and Chunliang Tao. Towards resilient and efficient llms: A comparative study of efficiency, performance, and adversarial robustness. *arXiv preprint arXiv:2408.04585*, 2024.
24. Xiaojing Fan, Chunliang Tao, and Jianyu Zhao. Advanced stock price prediction with xlstm-based models: Improving long-term forecasting. *Preprints*, (2024082109), August 2024.
25. Yixin Jin, Wenjing Zhou, Meiqi Wang, Meng Li, Xintao Li, Tianyu Hu, and Xingyuan Bu. Online learning of multiple tasks and their relationships: Testing on spam email data and eeg signals recorded in construction fields. *arXiv preprint arXiv:2406.18311*, 2024.
26. Panfeng Li, Youzuo Lin, and Emily Schultz-Fellenz. Contextual hourglass network for semantic segmentation of high resolution aerial imagery. In *2024 5th International Conference on Electronic Communication and Artificial Intelligence (ICECAI)*, pages 15–18. IEEE, 2024.
27. Dan Zhang, Fangfang Zhou, Felix Albu, Yuanzhou Wei, Xiao Yang, Yuan Gu, and Qiang Li. Unleashing the power of self-supervised image denoising: A comprehensive review. *arXiv preprint arXiv:2308.00247*, 2023.
28. Wanlong Liu, Shaohuan Cheng, Dingyi Zeng, and Qu Hong. Enhancing document-level event argument extraction with contextual clues and role relevance. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12908–12922, 2023.
29. Yiming Zhou, Zixuan Zeng, Andi Chen, Xiaofan Zhou, Haowei Ni, Shiyao Zhang, Panfeng Li, Liangxi Liu, Mengyao Zheng, and Xupeng Chen. Evaluating modern approaches in 3d scene reconstruction: Nerf vs gaussian-based methods. *arXiv preprint arXiv:2408.04268*, 2024.
30. Xinyu Shen, Qimin Zhang, Huili Zheng, and Weiwei Qi. Harnessing XGBoost for robust biomarker selection of obsessive-compulsive disorder (OCD) from adolescent brain cognitive development (ABCD) data. In Pier Paolo Piccaluga, Ahmed El-Hashash, and Xiangqian Guo, editors, *Fourth International Conference on Biomedicine and Bioinformatics Engineering (ICBBE 2024)*, volume 13252, page 132520U. International Society for Optics and Photonics, SPIE, 2024.
31. Huili Zheng, Qimin Zhang, Yiru Gong, Zheyan Liu, and Shaohan Chen. Identification of prognostic biomarkers for stage iii non-small cell lung carcinoma in female nonsmokers using machine learning. *arXiv preprint arXiv:2408.16068*, 2024.

32. Qimin Zhang, Weiwei Qi, Huili Zheng, and Xinyu Shen. Cu-net: a u-net architecture for efficient brain-tumor segmentation on brats 2019 dataset. *arXiv preprint arXiv:2406.13113*, 2024.
33. Letian Xu, Jiabei Liu, Haopeng Zhao, Tianyao Zheng, Tongzhou Jiang, and Lipeng Liu. Autonomous navigation of unmanned vehicle through deep reinforcement learning. *arXiv preprint arXiv:2407.18962*, 2024.
34. Liu Lipeng, Letian Xu, Jiabei Liu, Haopeng Zhao, Tongzhou Jiang, and Tianyao Zheng. Prioritized experience replay-based ddqn for unmanned vehicle path planning. *arXiv preprint arXiv:2406.17286*, 2024.
35. Hao Liu, Yi Shen, Chang Zhou, Yuelin Zou, Zijun Gao, and Qi Wang. Td3 based collision free motion planning for robot navigation. *arXiv preprint arXiv:2405.15460*, 2024.
36. Keqin Li, Jin Wang, Xubo Wu, Xirui Peng, Runmian Chang, Xiaoyu Deng, Yiwen Kang, Yue Yang, Fanghao Ni, and Bo Hong. Optimizing automated picking systems in warehouse robots using machine learning. *arXiv preprint arXiv:2408.16633*, 2024.
37. Keqin Li, Jiajing Chen, Denzhi Yu, Tao Dajun, Xinyu Qiu, Lian Jieting, Sun Baiwei, Zhang Shengyuan, Zhenyu Wan, Ran Ji, et al. Deep reinforcement learning-based obstacle avoidance for robot movement in warehouse environments. *arXiv preprint arXiv:2409.14972*, 2024.
38. Longsen Gao, Giovanni Cordova, Claus Danielson, and Rafael Fierro. Autonomous multi-robot servicing for spacecraft operation extension. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10729–10735. IEEE, 2023.
39. Zhizhong Wu. An efficient recommendation model based on knowledge graph attention-assisted network (kgatax). *arXiv preprint arXiv:2409.15315*, 2024.
40. Hao Yan, Zixiang Wang, Zhengjia Xu, Zhuoyue Wang, Zhizhong Wu, and Ranran Lyu. Research on image super-resolution reconstruction mechanism based on convolutional neural network, 2024.
41. Lianghao Tan, Shubing Liu, Jing Gao, Xiaoyi Liu, Linyue Chu, and Huangqi Jiang. Enhanced self-checkout system for retail based on improved yolov10. *arXiv preprint arXiv:2407.21308*, 2024.
42. Yue Guo, Shiqi Chen, Ronghui Zhan, Wei Wang, and Jun Zhang. Lmsd-yolo: A lightweight yolo algorithm for multi-scale sar ship detection. *Remote Sensing*, 14(19):4801, 2022.
43. Zhenglin Li, Bo Guan, Yuanzhou Wei, Yiming Zhou, Jingyu Zhang, and Jinxin Xu. Mapping new realities: Ground truth image creation with pix2pix image-to-image translation. *arXiv preprint arXiv:2404.19265*, 2024.
44. Wang Hu, Ashim Neupane, and Jay A. Farrell. Using PPP Information to Implement a Global Real-Time Virtual Network DGNSS Approach. *IEEE Trans. Veh. Technol.*, 71(10):10337–10349, 2022.
45. Panfeng Li, Qikai Yang, Xieming Geng, Wenjing Zhou, Zhicheng Ding, and Yi Nian. Exploring diverse methods in visual question answering. In *2024 5th International Conference on Electronic Communication and Artificial Intelligence (ICECAI)*, pages 681–685. IEEE, 2024.
46. Shicheng Liu and Minghui Zhu. Distributed inverse constrained reinforcement learning for multi-agent systems. *Advances in Neural Information Processing Systems*, 35:33444–33456, 2022.
47. Shicheng Liu and Minghui Zhu. Learning multi-agent behaviors from distributed and streaming demonstrations. *Advances in Neural Information Processing Systems*, 36, 2024.
48. Hanqian Li, Ruinan Zhang, Ye Pan, Junchi Ren, and Fei Shen. Lr-fpn: Enhancing remote sensing object detection with location refined feature pyramid network. *arXiv preprint arXiv:2404.01614*, 2024.
49. Haoqi Gao, Huafeng Wang, Zhou Feng, Mingxia Fu, Chennan Ma, Haixia Pan, Binshen Xu, and Ning Li. A novel texture extraction method for the sedimentary structures' classification of petroleum imaging logging. In *Pattern Recognition: 7th Chinese Conference, CCPR 2016, Chengdu, China, November 5-7, 2016, Proceedings, Part II 7*, pages 161–172. Springer, 2016.
50. Shicheng Liu and Minghui Zhu. Meta inverse constrained reinforcement learning: Convergence guarantee and generalization analysis. In *The Twelfth International Conference on Learning Representations*, 2023.
51. Jing Chen, Yan-Zhen Lu, Hao Jiang, Wei-Qing Lin, and Yong Xu. Twin model-based fault detection and tolerance approach for in-core self-powered neutron detectors. *Nuclear Science and Techniques*, 34(8):117, Aug 2023.
52. Wenbo Zhu. Optimizing distributed networking with big data scheduling and cloud computing. In *International Conference on Cloud Computing, Internet of Things, and Computer Applications (CICA 2022)*, volume 12303, pages 23–28. SPIE, 2022.
53. Yuqi Yan. Influencing factors of housing price in new york-analysis: Based on excel multi-regression model. 2022.

54. Tiechuan Hu, Wenbo Zhu, and Yuqi Yan. Artificial intelligence aspect of transportation analysis using large scale systems. In *Proceedings of the 2023 6th Artificial Intelligence and Cloud Computing Conference*, pages 54–59, 2023.
55. Xinjin Li, Yuanzhe Yang, Yixiao Yuan, Yu Ma, Yangchen Huang, and Haowei Ni. Intelligent vehicle classification system based on deep learning and multisensor fusion. In *Fifth International Conference on Computer Vision and Data Mining (ICCVDM 2024)*, volume 13272, pages 545–553. SPIE, 2024.
56. Yuchen Li, Haoyi Xiong, Linghe Kong, Qingzhong Wang, Shuaiqiang Wang, Guihai Chen, and Dawei Yin. S2phere: Semi-supervised pre-training for web search over heterogeneous learning to rank data. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4437–4448, 2023.
57. Yuchen Li, Haoyi Xiong, Qingzhong Wang, Linghe Kong, Hao Liu, Haifang Li, Jiang Bian, Shuaiqiang Wang, Guihai Chen, Dejing Dou, et al. Coltr: Semi-supervised learning to rank with co-training and over-parameterization for web search. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12542–12555, 2023.
58. Yuchen Li, Haoyi Xiong, Linghe Kong, Rui Zhang, Fanqin Xu, Guihai Chen, and Minglu Li. Mhrr: Moocs recommender service with meta hierarchical reinforced ranking. *IEEE Transactions on Services Computing*, 2023.
59. Yuanzhou Wei, Meiyan Gao, Jun Xiao, Chixu Liu, Yuanhao Tian, and Ya He. Research and implementation of cancer gene data classification based on deep learning. *Journal of Software Engineering and Applications*, 16(6):155–169, 2023.
60. Huili Zheng, Qimin Zhang, Yiru Gong, Zheyan Liu, and Shaohan Chen. Identification of prognostic biomarkers for stage iii non-small cell lung carcinoma in female nonsmokers using machine learning, 2024.
61. Xiaoyi Liu, Zhou Yu, and Lianghao Tan. Deep learning for lung disease classification using transfer learning and a customized cnn architecture with attention. *arXiv preprint arXiv:2408.13180*, 2024.
62. Haowei Ni, Shuchen Meng, Xieming Geng, Panfeng Li, Zhuoying Li, Xupeng Chen, Xiaotong Wang, and Shiyao Zhang. Time series modeling for heart rate prediction: From arima to transformers. *arXiv preprint arXiv:2406.12199*, 2024.
63. Haowei Ni, Shuchen Meng, Xupeng Chen, Ziqing Zhao, Andi Chen, Panfeng Li, Shiyao Zhang, Qifu Yin, Yuanqing Wang, and Yuxi Chan. Harnessing earnings reports for stock predictions: A qlora-enhanced llm approach. *arXiv preprint arXiv:2408.06634*, 2024.
64. Jing Chen, Ze-Shi Liu, Hao Jiang, Xi-Ren Miao, and Yong Xu. Anomaly detection of control rod drive mechanism using long short-term memory-based autoencoder and extreme gradient boosting. *Nuclear Science and Techniques*, 33(10):127, Oct 2022.
65. Jinglan Yang, Chaoqun Ma, Dengjia Li, and Jianghuai Liu. Mapping the knowledge on blockchain technology in the field of business and management: A bibliometric analysis. *IEEE Access*, 10:60585–60596, 2022.
66. Shisong Hsiao Jinglan Yang, Chaoqun Ma and Jianghuai Liu. Blockchain governance: a bibliometric study and content analysis. *Technology Analysis & Strategic Management*, 0(0):1–15, 2024.
67. Jinglan Yang, Jianghuai Liu, Zheng Yao, and Chaoqun Ma. Measuring digitalization capabilities using machine learning. *Research in International Business and Finance*, 70:102380, 2024.
68. Longsen Gao, Claus Danielson, and Rafael Fierro. Adaptive robot detumbling of a non-rigid satellite. *arXiv preprint arXiv:2407.17617*, 2024.
69. Yunzhi Fei, Yongxiu He, Fenkai Chen, Peipei You, and Hanbing Zhai. Optimal planning and design for sightseeing offshore island microgrids. In *E3S Web of Conferences*, volume 118, page 02044. EDP Sciences, 2019.
70. Baihe Gu, Hanbing Zhai, Yan An, Nguyen Quoc Khanh, and Ziyuan Ding. Low-carbon transition of southeast asian power systems–a swot analysis. *Sustainable Energy Technologies and Assessments*, 58:103361, 2023.
71. Chang Yu, Yixin Jin, Qianwen Xing, Ye Zhang, Shaobo Guo, and Shuchen Meng. Advanced user credit risk prediction model using lightgbm, xgboost and tabnet with smoteenn. *arXiv preprint arXiv:2408.03497*, 2024.
72. Yuchen Li, Haoyi Xiong, Linghe Kong, Zeyi Sun, Hongyang Chen, Shuaiqiang Wang, and Dawei Yin. Mpgraf: a modular and pre-trained graphformer for learning to rank at web-scale. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 339–348. IEEE, 2023.

73. Yuchen Li, Haoyi Xiong, Linghe Kong, Jiang Bian, Shuaiqiang Wang, Guihai Chen, and Dawei Yin. Gs2p: a generative pre-trained learning to rank model with over-parameterization for web-scale search. *Machine Learning*, pages 1–19, 2024.
74. Hanbing Zhai, Baihe Gu, Kaiwei Zhu, and Chen Huang. Feasibility analysis of achieving net-zero emissions in china's power sector before 2050 based on ideal available pathways. *Environmental Impact Assessment Review*, 98:106948, 2023.
75. Xinjin Li, Yu Ma, Yangchen Huang, Xingqi Wang, Yuzhen Lin, and Chenxi Zhang. Integrated optimization of large language models: Synergizing data utilization and compression techniques. *Preprints*, September 2024.
76. Wanyu Bian, Qingchao Zhang, Xiaojing Ye, and Yunmei Chen. A learnable variational model for joint multimodal mri reconstruction and synthesis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 354–364. Springer, 2022.
77. Wanyu Bian, Yunmei Chen, and Xiaojing Ye. Deep parallel mri reconstruction network without coil sensitivities. In *Machine Learning for Medical Image Reconstruction: Third International Workshop, MLMIR 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 8, 2020, Proceedings 3*, pages 17–26. Springer, 2020.
78. Wanyu Bian. *Optimization-Based Deep learning methods for Magnetic Resonance Imaging Reconstruction and Synthesis*. PhD thesis, University of Florida, 2022.
79. Qimin Zhang, Weiwei Qi, Huili Zheng, and Xinyu Shen. Cu-net: a u-net architecture for efficient brain-tumor segmentation on brats 2019 dataset, 2024.
80. Luoyao He, Xingqi Wang, Yuzhen Lin, Xinjin Li, Yu Ma, and Zhenglin Li. Boann: Bayesian-optimized attentive neural network for classification. 2024.
81. Zhizhong Wu. Mpgaan: Effective and efficient heterogeneous information network classification. *Journal of Computer Science and Technology Studies*, 6(4):08–16, 2024.
82. Xinhao Zhang, Zaitian Wang, Lu Jiang, Wanfu Gao, Pengfei Wang, and Kunpeng Liu. Tfwt: Tabular feature weighting with transformer. *arXiv preprint arXiv:2405.08403*, 2024.
83. Jinghan Zhang, Xiting Wang, Yiqiao Jin, Changyu Chen, Xinhao Zhang, and Kunpeng Liu. Prototypical reward network for data-efficient rlhf. *arXiv preprint arXiv:2406.06606*, 2024.
84. Yuxin Qiao, Keqin Li, Junhong Lin, Rong Wei, Chufeng Jiang, Yang Luo, and Haoyu Yang. Robust domain generalization for multi-modal object recognition. *arXiv preprint arXiv:2408.05831*, 2024.
85. LiMin Wang, XinHao Zhang, Kuo Li, and Shuai Zhang. Semi-supervised learning for k-dependence bayesian classifiers. *Applied Intelligence*, pages 1–19, 2022.
86. Zhuoyue Wang, Yiyi Tao, and Danqing Ma. A multiscale gradient fusion method for edge detection in color images utilizing the cbm3d filter. *arXiv preprint arXiv:2408.14013*, 2024.
87. Yiyi Tao, Zhuoyue Wang, Hang Zhang, and Lun Wang. Nevlp: Noise-robust framework for efficient vision-language pre-training. *arXiv preprint arXiv:2409.09582*, 2024.
88. Yiyi Tao, Yiling Jia, Nan Wang, and Hongning Wang. The fact: Taming latent factor models for explainability with factorization trees. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 295–304, 2019.
89. Yiyi Tao. Meta learning enabled adversarial defense. In *2023 IEEE International Conference on Sensors, Electronics and Computer Engineering (ICSECE)*, pages 1326–1330. IEEE, 2023.
90. Jiahao Tian, Jinman Zhao, Zhenkai Wang, and Zhicheng Ding. Mmrec: Llm based multi-modal recommender system. *arXiv preprint arXiv:2408.04211*, 2024.
91. Zhicheng Ding, Jiahao Tian, Zhenkai Wang, Jinman Zhao, and Siyang Li. Semantic understanding and data imputation using large language model to accelerate recommendation system. *arXiv preprint arXiv:2407.10078*, 2024.
92. Wu Xubo, Wu Ying, Li Xintao, Ye Zhi, Gu Xingxin, Wu Zhizhong, and Yang Yuanfang. Application of Adaptive Machine Learning Systems in Heterogeneous Data Environments. *Global Academic Frontiers*, 2(3), July 2024.
93. Han-Cheng Dan, Bingjie Lu, and Mengyu Li. Evaluation of asphalt pavement texture using multiview stereo reconstruction based on deep learning. *Construction and Building Materials*, 412:134837, 2024.
94. Han-Cheng Dan, Peng Yan, Jiawei Tan, Yinchao Zhou, and Bingjie Lu. Multiple distresses detection for asphalt pavement using improved you only look once algorithm based on convolutional neural network. *International Journal of Pavement Engineering*, 25(1):2308169, 2024.

 doi:10.20944/preprints202410.2397.v1

95. Yiru Gong, Qimin Zhang, Huili Zheng, Zheyan Liu, and Shaohan Chen. Graphical Structural Learning of rs-fMRI data in Heavy Smokers. *arXiv preprint arXiv:2409.08395*, 2024.
96. Xiaoyi Liu, Zhou Yu, Lianghao Tan, Yafeng Yan, and Ge Shi. Enhancing skin lesion diagnosis with ensemble learning. *arXiv preprint arXiv:2409.04381*, 2024.
97. Zixuan Wang, Yanlin Chen, Feiyang Wang, and Qiaozhi Bao. Improved unet model for brain tumor image segmentation based on aspp-coordinate attention mechanism. *arXiv preprint arXiv:2409.08588*, 2024.
98. Zhizhong Wu, Xueshe Wang, Shuaishuai Huang, Haowei Yang, Danqing Ma, et al. Research on prediction recommendation system based on improved markov model. *Advances in Computer, Signals and Systems*, 8(5):87–97, 2024.
99. Zhizhong Wu. Deep learning with improved metaheuristic optimization for traffic flow prediction. *Journal of Computer Science and Technology Studies*, 6(4):47–53, 2024.
100. Zheng Lin, Chenghao Wang, Zichao Li, Zhuoyue Wang, Xinqi Liu, and Yue Zhu. Neural radiance fields convert 2d to 3d texture. *Applied Science and Biotechnology Journal for Advanced Research*, 3(3).