Hindawi
International Journal of Computer Games Technology
Volume 2023, Article ID 3368647, 11 pages
https://doi.org/10.1155/2023/3368647

Research Article

Deep Generative Modeling Based on VAE-GAN for 3D Indoor Scene Synthesis

Shuai Li and Hongjun Li

College of Science, Beijing Forestry University, Beijing 100083, China

Correspondence should be addressed to Hongjun Li; lihongjun69@bjfu.edu.cn

Received 23 December 2022; Revised 25 August 2023; Accepted 5 September 2023; Published 20 September 2023

Academic Editor: Davide Gadia

With the advancement of virtual reality and 3D game technology, the demand for high-quality 3D indoor scene generation has surged. Addressing this need, this paper presents a method leveraging a VAE-GAN-based framework to conquer two primary challenges in 3D scene representation and deep generative networks. First, we introduce a matrix representation to encode fine-grained object attributes, alongside a complete graph to implicitly capture object spatial relations—effectively encapsulating both local and global scene structures. Second, we devise a unique generative framework based on VAE-GAN and the Bayesian optimization. This framework learns a Gaussian distribution of encoded object attributes through a VAE-GAN network, allowing for sampling and decoding of the distribution to generate new object attributes. Subsequently, a U-Net is employed to learn spatial relations between objects. Lastly, the Bayesian optimization module amalgamates the generated object attributes, spatial relations, and priors learned from data, conducting global optimization to generate a logical scene layout. Experimental results on a large-scale 3D indoor scene dataset substantiate that our method effectively learns inter-object relations and generates diverse and plausible indoor scenes. Comparative experiments and user studies further validate that our method surpasses the current state-of-the-art techniques in indoor scene generation and is comparable to real training scenes.

1. Introduction

The comprehension and creation of 3D models serve as vital challenges within the fields of computer graphics and vision, with a particular emphasis on 3D indoor scene generation. Technological advancements in robotics, virtual reality, smart homes, and 3D gaming have surged the need for high-quality virtual 3D indoor scenes. Concurrently, strategies for analyzing and modeling 3D models are increasingly shifting towards data-driven learning techniques. Furthermore, the evolution of 3D scanning technologies, combined with a decrease in cost for scanning devices such as lidars and Kinect, has drastically reduced the expense of obtaining 3D data. Recent research demonstrates that generative neural networks are adept at producing high-quality images, speech, and 3D shapes. These progresses have made deep-learning-based 3D indoor scene generation a tangible reality.

3D indoor scene generation presents two principal challenges, the first of which is the issue of scene representation. Unlike isolated 3D shapes, 3D indoor scenes display substantial variation in terms of object categories, shapes, positions, and orientations, leading to intricate structures. The scene representation must account for a wide array of patterns, encompassing continuous factors like object sizes and relative positions, as well as discrete elements such as object adjacencies, symmetries, and cooccurrences. Moreover, it should accommodate geometric constraints like interpenetration avoidance and support relations. Various methods, including top-view representations [1, 2], hierarchical graph structures [3], and hybrid representations [4], have been developed to tackle this. While top-view-based methods can depict adjacency relations between objects, they concentrate solely on local information in the projection plane and fail to illustrate the scene's hierarchical structure. Despite the ability of graph-based representations to convey object hierarchies, they necessitate the predefinition of various relations, making it challenging to express implicit relations and thus somewhat curtailing their representational power. In

contrast, our approach entails representing the object set of a scene as a matrix, with each column vector encoding an object. We use a parameterized complete graph to express the spatial relations between objects, which implicitly represents both local and global spatial relations without necessitating additional local or global supervision. The benefit of this comprehensive relational representation is its ability to optimize all factors in scene generation concurrently, a feat not easily achieved by recursive methods due to their difficulty in recovering from errors.

Another formidable challenge in 3D indoor scene generation is crafting suitable neural networks that can capture the intricate geometric structures and spatial relations within a scene. Neural networks possess boundless expressive capacity, allowing them to effectively encode both continuous and discrete scene patterns and learn scene priors that are not manually inserted by human designers. Certain methodologies [4–6] have endeavored to apply successful generative models such as variational autoencoders (VAE) [7, 8] and generative adversarial networks (GAN) [9–12] to the 3D domain. Despite these models marking significant advancements in 3D model generation, issues such as mode collapse and the generation of blurry results persist. VAE, with its structured continuous latent space, is less susceptible to mode collapse but often produces blurry samples. Conversely, GAN, despite its latent space being insufficiently structured and continuous which could lead to mode collapse, is capable of efficiently generating realistic samples. To address this, our paper merges the strengths of both models, proposing a novel 3D indoor scene generation method founded on VAE-GAN and the Bayesian optimization. Specifically, we first encode the object attributes of the scene via VAE-GAN and generate new object attributes by sampling and decoding the latent space. These new attributes are subsequently processed by a U-Net to learn spatial relational attributes. Deeper networks are utilized to learn global relations, a key aspect of indoor scene generation. Ultimately, the new object attributes, spatial relational attributes, and the priors learned from the input data are collectively fed into a Bayesian scene optimization framework for further global optimization, leading to the generation of realistic new scenes.

We conducted a comprehensive evaluation of our method on the large-scale 3D indoor scene dataset 3D-FRONT [13]. We tested the generation performance on two room types—bedroom and living room. Comparisons were made against two state-of-the-art methods. The results show that our method is capable of generating realistic and diverse indoor scenes, outperforming the previous advanced methods.

In summary, our main contributions are three-fold: (1) We introduced a matrix and compete graph scene representation that can effectively encode object attributes, as well as local and global interobject relations. (2) We proposed a novel scene generation framework based on VAE-GAN and the Bayesian optimization, which reduces blurriness and improves quality and realism of the generated 3D scenes. (3) We conducted comprehensive experiments on a large-scale 3D indoor scene dataset, validating the efficacy of our method and demonstrating performance surpassing current state-of-the-art techniques.

2. Related Work

Our work proposes to design a deep generative neural network for 3D indoor scene modeling. As such, the related work section focuses primarily on previous methods for 3D modeling and synthesis.

2.1. 3D Representation. 3D object and scene representations form the crux of various fields such as computer vision, robotics, augmented reality, and virtual reality. Unlike the natural vector representations of images and videos, parametric encoding of 3D geometry provides a high degree of flexibility. Over the past decades, AI and vision researchers have proposed various 3D model representations, including multiview [14–16], voxel [17–19], point cloud [20], part-based [21, 22], graph/mesh [23, 24], and spherical [25, 26] representations.

Contemporary research focused on creating parameterized 3D model representations has primarily targeted 3D shapes. For instance, Wu et al. [27] put forth 3D-GAN, a voxel-based 3D shape generation network, which exhibited promising results in shape generation and reconstruction. However, voxel representations grapple with resolution limitations that result in the loss of local details, rendering their extension to 3D indoor scene modeling and generation challenging. Tulsiani et al. [22] suggested a part-based model that assembles shapes using 3D voxel primitives. Similarly, Nash and Williams [28] introduced ShapeVAE, employing an encoder-decoder structure to generate 3D point clouds with semantic labels.

These aforementioned methods are specifically tailored to the characteristics of 3D shapes, and their extension to 3D indoor scenes presents inherent challenges. The variations in indoor scenes are substantially more complex than those in 3D shapes, and the spatial relations prove to be difficult for shape generation models to capture. Methods that are effective on shapes often do not replicate the same efficiency on scenes. Such challenges have spurred us to conceive new specialized representations and training schemes specifically designed for 3D indoor scene generation.

2.2. 3D Indoor Scene Synthesis. Indoor scene modeling stands as a vital component of 3D content creation. The prevalent strategies for automatically synthesizing virtual 3D indoor scenes include probability modeling centered around assembly, graph models, and deep neural network-based methods.

Historically, an array of early research utilized probabilistic graphical models, such as the Bayesian networks, to facilitate assembly-based modeling and synthesis [29, 30]. Fisher et al. [31] devised a probabilistic model for scenes, built upon the Bayesian networks and the Gaussian mixture models, thereby addressing challenges related to object appearance and layout optimization in subscene modeling. This approach harnessed subscenes to glean prior knowledge for synthesizing novel scenes. Concurrently, graph models have been employed to encapsulate the overarching layout structure of scenes. In their work, Kermani et al. [32] treated the scene as a graph model, adhering to

cooccurrence and permutation models, and progressively integrated objects into the scene, synthesizing 3D indoor scenes. In a unique approach, Wang et al. [33] proposed a method that melds relational graphs with top-view, bifurcating scene generation into planning and instantiation stages.

In more recent years, deep neural networks have gained widespread adoption for 3D indoor scene generation. Wang et al. [1] pioneered learning convolutional priors and proposed an indoor scene synthesis approach rooted in convolutional neural networks. They utilized semantically rich orthogonal top-view images to encode scene composition and layout and trained convolutional networks to produce a two-dimensional distribution of object placement. In an innovative step, Li et al. [3] became the first to utilize variational autoencoders (VAEs) for recursively learning the support and cooccurrence relationships within scenes, portraying the scene as a hierarchical tree structure and deploying encoders for object grouping and decoders for scene generation. Zhang et al. [4] suggested a hybrid method of matrix and 2D image representation, employing a VAE-GAN model for training and generating fresh indoor scenes.

In our methodology, we also leverage a matrix to represent the object arrangement in indoor scenes. However, to depict the spatial relationships between objects, we employ a parameterized complete graph representation, eschewing an image representation that lacks spatial information.

2.3. Deep Generative Models for 3D Modeling. The surge of online visual data repositories like ImageNet [34] and ShapeNet [35] has catalyzed a significant focus within computer vision: learning parametric models from vast datasets to capture subtle shape variations in geometric data. Initial parametric learning models were predominantly trained on faces and bodies [36–38], using deformation of templates for modeling. However, these approaches are only fitting for objects with minor geometric and topological alterations, making them unsuitable for 3D indoor scenes that display substantial variations.

The remarkable success of deep neural networks in recent years has ushered in new opportunities for data-driven 3D modeling. The vision community is now concentrated on utilizing neural networks to encode mapping functions, as demonstrated in models such as generative adversarial networks (GANs) [9–12], variational autoencoders (VAEs) [7, 8], and autoregressive models [39]. These models are proficient at modeling high-dimensional data and can generate new samples by sampling from lower-dimensional spaces. Despite their impressive performance on 2D images, scaling these techniques to accommodate 3D data continues to pose challenges.

3. Indoor 3D Scene Representation and Generation

3.1. Overview. Our scene generation framework, based on the VAE-GAN model, is trained using the extensive 3D indoor scene dataset 3D-FRONT [13]. Each scene in this dataset is presegmented into objects across various categories, with each object defined by a category label, size, pose, and shape encoding. After training, our framework can generate new indoor scenes by decoding randomly sampled noise vectors and replacing the shape encodings with retrieved 3D object models.

3D indoor scene representation necessitates the capture of both the continuous patterns of object attributes and the discrete patterns of spatial relations between objects. Consequently, the quality of scene generation is intrinsically tied to the effectiveness of scene representation. To encode object attributes, we employ a matrix representation for the set of objects in a scene, a technique reminiscent of approaches used in references [4, 5]. Each column vector within this matrix embodies the attribute encoding of an object along with a binary indicator delineating the selection status of the object category. For encoding spatial relations, our method diverges from the one in reference [3], eliminating the requirement for predefined interobject relations. Instead, we extract relative attributes using a U-Net, enabling us to parameterize spatial relations between objects. In every newly generated scene, a VAE-GAN is utilized to create the attributes of objects. The synthesized relative attributes are then produced by inputting these attributes into the U-Net, thereby providing an overcomplete constraint set for the final object attributes. This process permits the restoration of accurate attributes through consistency constraints, regardless of any potential erroneous predictions.

To further refine the results generated by the VAE-GAN, we learn the prior distributions of both object attributes and relative attributes, thereby providing a measure of prediction uncertainty. This approach encompasses both continuous variables such as relative positions, and discrete variables like object counts and cooccurrences, thereby enhancing the regularization of the final output. Ultimately, a Bayesian scene optimization framework amalgamates these neural predictions and prior distributions to produce the final object attributes.

3.2. 3D Scene Representation. To enable a viable representation of the 3D indoor scene, denoted as S, we employ a parametric approach. This results in each 3D indoor scene being represented as a matrix, M_S, as illustrated in Figure 1. We represent the collection of objects within the indoor scene as set O. Each object code, denoted as column v^o within the matrix M_S, corresponds to an object $o \epsilon O$ present in the scene. Assuming a total of N_c object categories, each object in the scene belongs to a specific category $k \epsilon C$. Given the possibility of multiple identical object categories within a scene—for instance, multiple chairs around a dining table or several bedside tables around a bed—we further assume that a scene can include up to m_k objects from each category k. Consequently, the total number of objects within each scene is bounded by $n = \sum_{k=1}^{N_c} m_k$. In our experimental setup, we set $n_c = 20$ and $m_k = 4$, which implies that each scene can contain a maximum of 80 objects.

In terms of object attributes, we represent each object $o \epsilon O$ within the scene using a state vector $v^o \epsilon R^{10}$. The components of this vector are as follows. $s_o = (v_0^o, v_1^o, v_2^o)^T$ encodes the scale of the object o, aligned with the axis of the

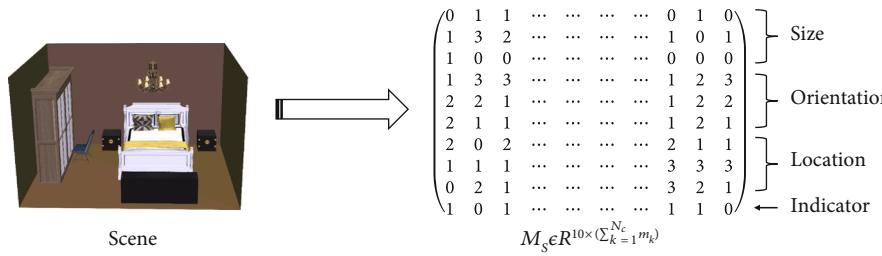

Figure 1: Representation of indoor scene using matrix.

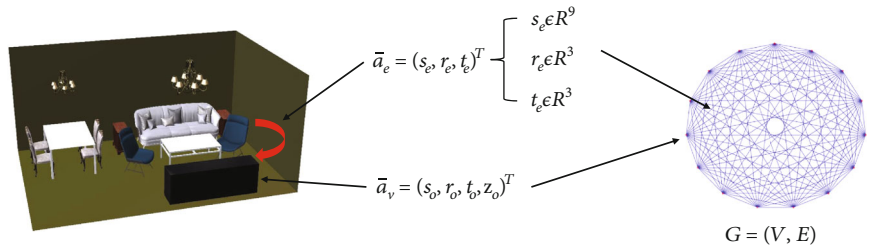

Figure 2: Spatial relationship graph representation of indoor 3D scenes.

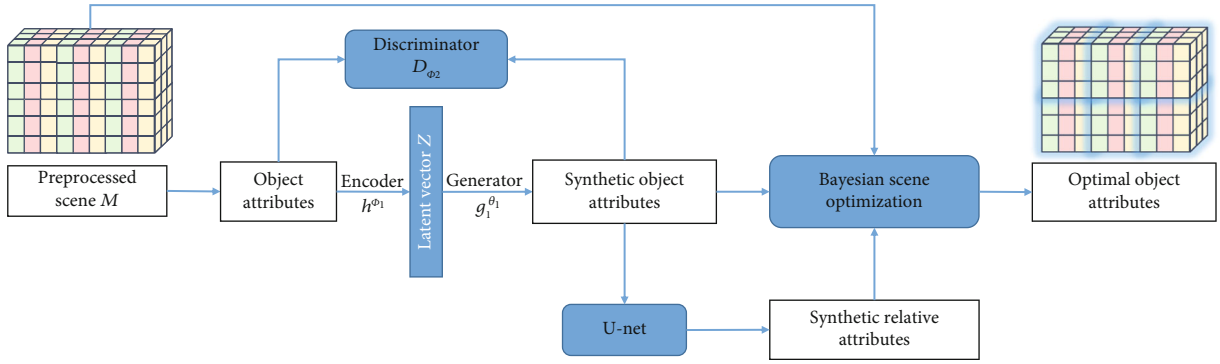

Figure 3: The structure of 3D scene synthesis network.

coordinate system associated with each object. $r_o = (v_3^o, v_4^o, v_5^o)^T$ encodes the Euler angles of the orientation of each object o in the world coordinate system. $t_o = (v_6^o, v_7^o, v_8^o)^T$ encodes the position of each object o in the world coordinate system. $z_o = v_9^o$ acts as an object presence indicator, specifying whether the object o is present in the scene.

As a result, the set of objects in a 3D indoor scene can be parameterized as a matrix $M_S \epsilon R^{10\times(\sum_{k=1}^{N_c} m_k)}$. Each column of the matrix, denoted as A_o, represents an object attribute encoding $(s_o, r_o, t_o, z_o)^T$.

Relative attributes capture spatial relationships such as adjacency, support, symmetry, and others between pairs of objects within a scene. Unlike predefined explicit representations, we also parametrically express these object relationships implicitly. For relative object attributes, we employ the method described in [5] to represent these properties. We visualize a 3D scene as a complete graph $G = (V, E)$, as depicted in Figure 2. Each vertex $v \epsilon V$ represents an object in the scene, while the edges $e \epsilon E$ signify the connections between objects. Each edge $e \epsilon (v, v')$ is encoded as the associated attribute $a_e = (s_e, r_e, t_e)^T$, where $s_e \epsilon R^9$ represents the pairwise differences between s_v and $s_{v'}$ for the three scale parameters. $r_e \epsilon R^3$ denotes the Eulerian angle of the pose of v' in the local coordinate system of v; $t_e \epsilon R^3$ denotes the center position of v' in the local coordinate system of v.

Therefore, the entire set of a scene's relative attributes can be represented by a tensor A_ε, i.e.

$$A_\varepsilon \epsilon R^{15\times\left(\sum_{k=1}^{N_c} m_k\right)\times\left(\sum_{k=1}^{N_c} m_k\right)}. \quad (1)$$

3.3. 3D Scene Synthesis Network. Our network architecture, depicted in Figure 3, leverages the VAE-GAN framework described in [40] to synthesize object attributes. This architecture combines the high-quality generative capabilities of GANs with the efficient data encoding into latent space offered by VAEs, facilitating the generation of samples conditioned on prior data. Our network comprises two primary

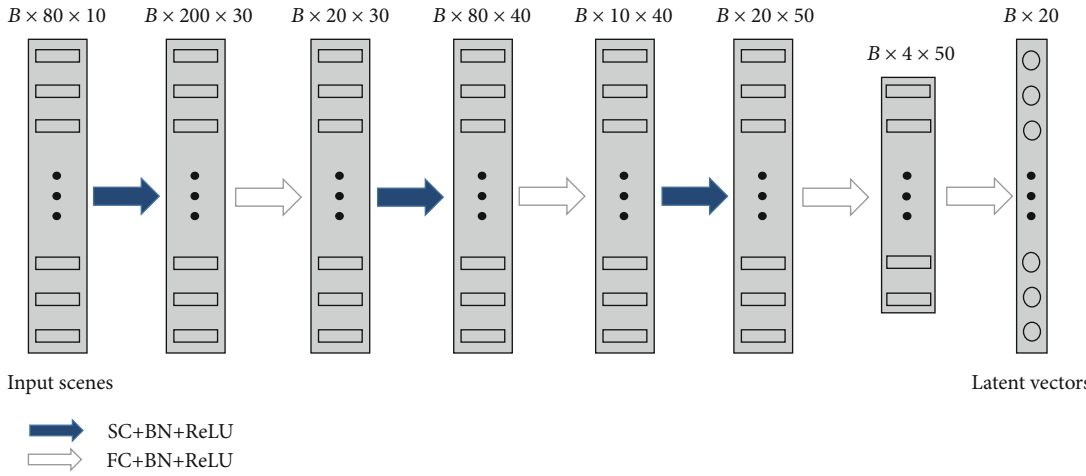

Figure 4: Encoder structure.

components: a complete VAE and a GAN, which share a decoder, parameters, and a simultaneous training regimen. The first part of our network, the encoder, learns the latent space of the data from the input. The generator then samples from this latent space to create novel object attributes, attempting to fool the discriminator. Tasked with distinguishing between real and synthesis object attributes, the discriminator serves as a critic in this architecture. As illustrated in Figure 4, the encoder within our VAE-GAN model boasts a unique design, mirrored in the architecture of the generator. We have designed our network to mitigate overfitting and enhance generalizability by using sparsely connected layers as an alternative to fully connected ones. The layers in our network module alternate between sparse and full connections. For improved training stability, we incorporate batch normalization and ReLU layers between each layer. The encoder's latent space, once generated, is sampled to synthesize the attributes of objects in a new 3D scene via the generator. The output from the VAE-GAN network is then fed into a separate U-net network, which generates the relative attributes of the objects' spatial relationships in the new scene.

In the VAE-GAN network, we refer to the encoder, generator, and discriminator as h^{Φ_1}, $g_1^{\theta_1}$, and D_{Φ_2}, respectively. The U-Net module, responsible for the relative attributes of objects, is denoted as $g_2^{\theta_2}$. Here, $\varnothing = (\Phi_1, \Phi_2)$, and $\theta = (\theta_1, \theta_2)$ represent the network parameters. The training set is denoted as $T = \{(A_v, A_\varepsilon)\}$, where A_v and A_ε correspond to the encoded object attributes and relative attributes, respectively.

Similar to the VAE-GAN setup, our loss function is composed of three components: the reconstruction loss L_{recon}, the discriminator loss L_D, and the KL divergence loss L_{KL}. The latter is used to constrain the distribution of the encoder's output. Therefore, the total loss can be expressed as follows:

$$L = L_{\text{recon}} + \mu_\varepsilon L_D + \lambda_{Kl} L_{KL}, \tag{2}$$

where μ_ε and λ_{Kl} are the weights of discriminator loss and KL divergence loss, respectively.

$$L_{\text{recon}} = \frac{1}{|T|} \sum_{(A_v, A_\varepsilon) \in T} (\lambda_\varepsilon f(A_v, A_\varepsilon) + g(A_v)), \tag{3}$$

where

$$\begin{aligned} f(A_v, A_\varepsilon) &= \left\| g_2^{\theta_2}\left(g_1^{\theta_1}\left(h^{\Phi_1}\left(\bar{A}_v\right)\right)\right) - \bar{A}_\varepsilon \right\|^2, \\ g(A_v) &= \left\| g_1^{\theta_1}\left(h^{\Phi_1}\left(\bar{A}_v\right)\right) - \bar{A}_v \right\|^2, \\ L_D &= \frac{1}{|T|} \sum_{(A_v, A_\varepsilon) \in T} D_{\Phi_2}\left(\bar{A}_v\right) - E_{z \sim N_d} D_{\Phi_2}\left[g_1^{\theta_1}(z)\right], \\ L_{\text{KL}} &= \text{KL}\left(\left\{h^{\Phi_1}\left(\bar{A}_v\right)\right\} | N_d\right). \end{aligned} \tag{4}$$

Ultimately, we learn the modules in the network architecture by optimizing the following objective function:

$$\begin{aligned} &\min_{\Phi_1, \theta} \max_{\Phi_2} \frac{1}{|T|} \sum_{(A_v, A_\varepsilon) \in T} (\lambda_\varepsilon f(A_v, A_\varepsilon) + g(A_v)) \\ &+ \mu_\varepsilon \left(\frac{1}{|T|} \sum_{(A_v, A_\varepsilon) \in T} D_{\Phi_2}\left(\bar{A}_v\right) - E_{z \sim N_d} D_{\Phi_2}\left[g_1^{\theta_1}(z)\right] \right) \\ &+ \lambda_{Kl} \text{KL}\left(\left\{h^{\Phi_1}\left(\bar{A}_v\right)\right\} | N_d\right). \end{aligned} \tag{5}$$

This function integrates the autoencoder (AE) loss and the discriminator loss from the GAN. Both of these losses are defined on latent variables, with the discriminator loss penalizing discrepancies between scenes generated by the generator and the corresponding input scenes. The latent distribution N_d is a standard normal distribution. KL represents the Kullback–Leibler divergence loss term. The discriminator D_Φ shares the same network structure as the

generator $g_1^{\theta_1}$, expect for the use of a single value of the latent vector. In this study, we set $\lambda_\varepsilon = 1$, $\mu_\varepsilon = 1$, and $\lambda_{\mathrm{KL}} = 0.01$.

Considering the highly nonconvex nature of the aforementioned objective function, its direct optimal solution poses a considerable challenge. In this study, we adopt the alternating minimization method. This approach decomposes the objective function into two subproblems, each easier to optimize. Specifically, we proceed as follows.

3.3.1. Generator Optimization. With Φ_2 held constant, the optimization problem simplifies as follows:

$$\min_{\Phi_1,\theta} \frac{1}{|T|} \sum_{(A_v,A_\varepsilon)\in T} (\lambda_\varepsilon f(A_v, A_\varepsilon) + g(A_v)) - E_{z\sim N_d} D_{\Phi_2}\left[g_1^{\theta_1}(z)\right]. \tag{6}$$

We employ the ADAM optimizer [41] to address this problem, using a learning rate of 0.001.

3.3.2. Discriminator Optimization. When Φ_1 and θ are held constant, the aforementioned optimization problem reduces to

$$\min_{\Phi_2} -\left(\frac{1}{|T|} \sum_{(A_v,A_\varepsilon)\in T} D_{\Phi_2}(\bar{A}_v) - E_{z\sim N_d} D_{\Phi_2}\left[g_1^{\theta_1}(z)\right]\right). \tag{7}$$

Once again, we utilize the ADAM optimizer [41] to optimize the aforementioned problem, adopting a learning rate of 0.001.

3.3.3. Bayesian Optimization of Scene. Neural networks have robust expressive capabilities and can learn scene design priors from large-scale indoor scene datasets, a feat typically beyond the scope of human design. To further optimize the generated scenes and achieve superior results, we learn the prior distribution of the data from the training set. We then integrate these priors with the neural network's predictions (object attributes, relative attributes) using the Bayesian framework described in [5]. Let $\bar{a}_v = (s_o, r_o, t_o, z_o)^T$, $\bar{a}_v^0$, and $\bar{a}_e^0$ denote the vertex edge predictions from the VAE-GAN and U-Net neural network outputs, respectively. The input for the Bayesian scene optimization consists of the prediction $\bar{a}_v^0$ associated with each vertex $v \epsilon V$ and the prediction $\bar{a}_e^0$ associated with each edge $e = (v, v')^T \epsilon E$. We then formulate the scene optimization as a posterior distribution maximization problem:

$$P(\{\bar{a}_v\}|\{\bar{a}_v^0\} \cup \{\bar{a}_e^0\}) \sim P(\{\bar{a}_v^0\} \cup \{\bar{a}_e^0\}|\{\bar{a}_v\}) \bullet P(\{\bar{a}_v\}). \tag{8}$$

Here, $P(\{\bar{a}_v^0\} \cup \{\bar{a}_e^0\}|\{\bar{a}_v\})$ and $P(\{\bar{a}_v\})$ represent the total likelihood term and the prior term, respectively. The symbol ~ denotes proportionality. For the solution to this posterior maximization problem, please refer to [5].

4. Experiment

Our experiments were performed on a desktop with an NVIDIA GeForce 1070 GPU, an Intel Xeon(R) E5-2640 v3 @2.60GHz CPU with 16 cores, and 32GB of memory. The training stage of the scene generation model rans on the GPU, with the training time depending on the size of the dataset and the complexity of the scenes.

4.1. Dataset and Preprocessing. Over the past few years, there have been significant advancements in neural network-based, data-driven indoor scene generation methodologies, such as those reported in [1–4]. These works were predominantly implemented on the SUNCG [42] dataset, which is now unavailable due to legal concerns and differs from the 3D-FRONT dataset [13] in several ways. To provide a fair comparison, we selected state-of-the-art 3D indoor scene synthesis models Sync2Gen [5] and FastSynth [2] as our benchmarks on the 3D-FRONT dataset. It is worth noting that FastSynth was initially evaluated on the SUNCG dataset, so we made some modifications based on the author's implementation for training and evaluation on 3D-FRONT.

This extensive synthetic indoor scene dataset covers a broad range of scene types, such as bedrooms, living rooms, kitchens, and offices, and encompasses 18,797 rooms replete with high-quality textured 3D objects. In keeping with previous studies [1–4, 33], for comparability, we extracted two types of scene data from the dataset—bedrooms and living rooms. These categories are predominant in residential indoor scenes and represent the largest categories within the dataset, making them our choice for the training set. The bedroom category includes three subtypes: bedroom, master bedroom, and second bedroom, while the living room category comprises two subtypes: living room and living dining room.

To ensure synthesized scenes were more type-specific, given the comprehensive size of the 3D-FRONT dataset, we performed some preprocessing. This entailed extracting the 20 most frequently appearing object categories in each room type, thereby omitting the least common object categories. We excluded scenes with fewer than six objects, excessively large dimensions (length and width exceeding 8 m), or rooms containing more than four objects per category. For bedroom-type rooms, we specifically omitted data from rooms devoid of beds. Following this preprocessing, we acquired 3,397 bedroom-type rooms (3,000 for training, 397 for validation) and 4,893 living room-type rooms (4,000 for training, 893 for validation). The distribution of these statistics is detailed in Table 1.

4.2. Evaluation Metrics. To evaluate the realism of the scenes generated by our approach, we follow the evaluating methodology adopted in [2]. The main idea of the evaluating methodology is to use machine learning methods to distinguish images. If a trained binary classifier cannot accurately distinguish whether an image is a synthesized image or a real photo, it indicates that the synthesized image has a good sense of realism and can be confused with fake images. Therefore, we in advance train a binary classifier to

Table 1: Experimental data statistics table.

Bedroom	Bedroom		Living room	
	Master bedroom	Second bedroom	Living dining room	Living room
3323	4526	3485	4844	1848
	Train	Validation	Train	Validation
	3000	397	4000	893

Figure 5: Eight randomly generated bedroom scenes, showcasing variability in object count and layout design.

distinguish between synthesized and real scenes, reporting the resulting classification accuracy. Specifically, this classifier, same as depicted in Figure 4, integrates an encoder framework. However, in place of the latent vector, a binary classification head is employed. This structure takes the same matrix representation leveraged by our model. Training of the classifier is undertaken on 2,000 scenes, evenly split between real and synthesized scenes. We then evaluate the classifier's accuracy on 200 synthesized scenes. It is worth highlighting that this classifier bears similarities to the discriminator component in generative adversarial networks (GANs). In scenarios where the generated scenes are closely the real ones, the classifier grapples with distinguishing between the two. Consequently, an accuracy rate approximating 50% is indicative of superior performance—the closer to this benchmark, the better the quality of the synthesized scenes.

Additionally, we also evaluate the Kullback–Leibler (KL) divergence also adopted in [2] between the distributions of synthesized and real scenes, providing a measure of their similarity. This metric offers insights into their similarity, with a lower value being more desirable.

Moreover, we undertake a perceptual study involving nonexpert users, contrasting rendered views of scenes generated by our approach against baseline and ground truth scenes. Specifically, we present 60 pairs of scene view images to 30 participants in a questionnaire, soliciting their choices for the scene with the more plausible layout. Critically, we utilize the same object category models for rendering all scene types, ensuring that we eliminate potential biases from factors such as object shapes and material appearances. Lastly, we quantify the results by calculating the percentage of cases in which scenes generated by our method are deemed to exhibit more appealing and logical layouts than those of the baseline and ground truth scenes—the higher the percentage, the better the performance.

4.3. Experiment Results. We conducted 1000 iterations of training on both scene types, which took 34 hours for bedroom scenes and 44 hours for the more intricate living room scenes. The testing and optimization stages ran on the CPU, averaging around 5 seconds to generate a new scene.

4.3.1. Scene Generation. Figures 5 and 6 showcase examples of the two scene types synthesized by our method. The generated 3D indoor scenes demonstrate satisfactory and plausible layouts, encompassing a variety of object types. As illustrated in the figures, we can observe that objects within both room types exhibit excellent adjacency and spatial relationships. For example, nightstands are positioned on either side of the bed; wardrobes are usually placed against the wall and located close to the bed in bedroom scenes; TV stands typically face the bed or the sofa, and so on. Simultaneously, we can see that pendant lamps are found overhead in the overall scene, usually situated above the bed or dining table. This suggests that our network can successfully learn not only adjacency relations but also hierarchical spatial relations among objects in the scene.

As depicted in Figure 6, we notice that dining chairs are arranged around dining tables and can be tucked under them, underscoring our network's ability to grasp embedding relations among objects. These visualizations confirm

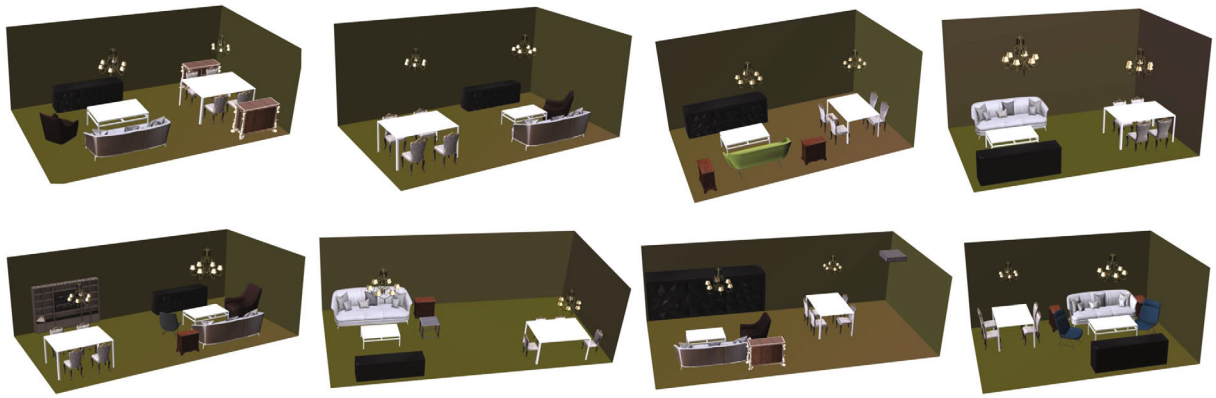

Figure 6: Eight randomly generated living room scenes, showcasing variability in object count and layout design.

Table 2: Quantitative comparisons of scene synthesis.

	Scene classification accuracy (↓)			Category KL divergence (↓)		
	Sync2Gen	FastSynth	Ours	Sync2Gen	FastSynth	Ours
Bedroom	0.862	0.896	0.753	0.0064	0.0085	0.0056
Living room	0.915	0.963	0.825	0.0188	0.0399	0.0272

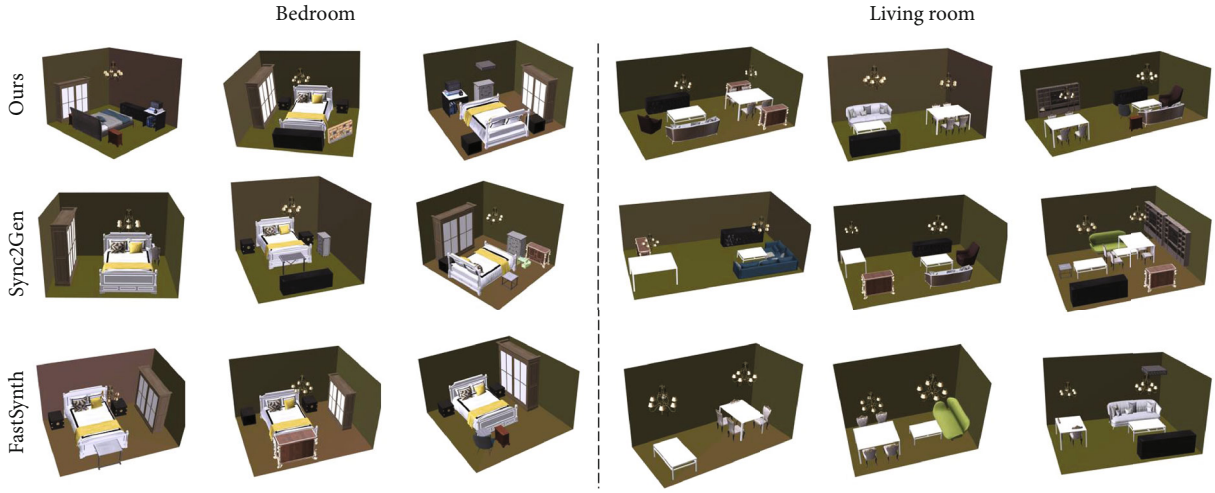

Figure 7: Visual comparison of different methods. The first row represents scenes generated by our method, the second row illustrates the Sync2Gen method, and the third row displays the FastSynth method.

that our neural network is adept at generating sensible and appealing room layouts for both simpler bedroom scenes and more complex living room scenes. It is capable of learning adjacency, spatial relations, and embedding relations among objects in the scenes.

4.3.2. Realism Evaluation. In addition, to evaluate the realism of the samples generated by our method quantitatively, we trained a binary classifier to measure the similarity between our synthetic samples and the real samples present in the 3D-FRONT dataset. As depicted in Table 2, our method's classification accuracy is consistently closer to 50% for both scene types compared to the other two baseline methods. This indicates that the samples generated by our model closely resemble real scenes, making it challenging for the classifier to distinguish them, hence suggesting higher realism.

We also report the KL divergence between the distributions of samples generated by our method and real samples. Our method consistently demonstrates lower KL divergence, indicating that our model generates samples that closely mirror the training set's distribution, which further attests to the realism of the scenes generated by our model. The visual comparison results are depicted in Figure 7. Both baseline methods exhibit suboptimal layout designs, whereas our method can generate more realistic and reasonable scenes.

4.3.3. Diversity Evaluation. The scenes generated by our method are diverse. Figure 8 displays these generated scenes alongside the most similar scenes from the training set. The

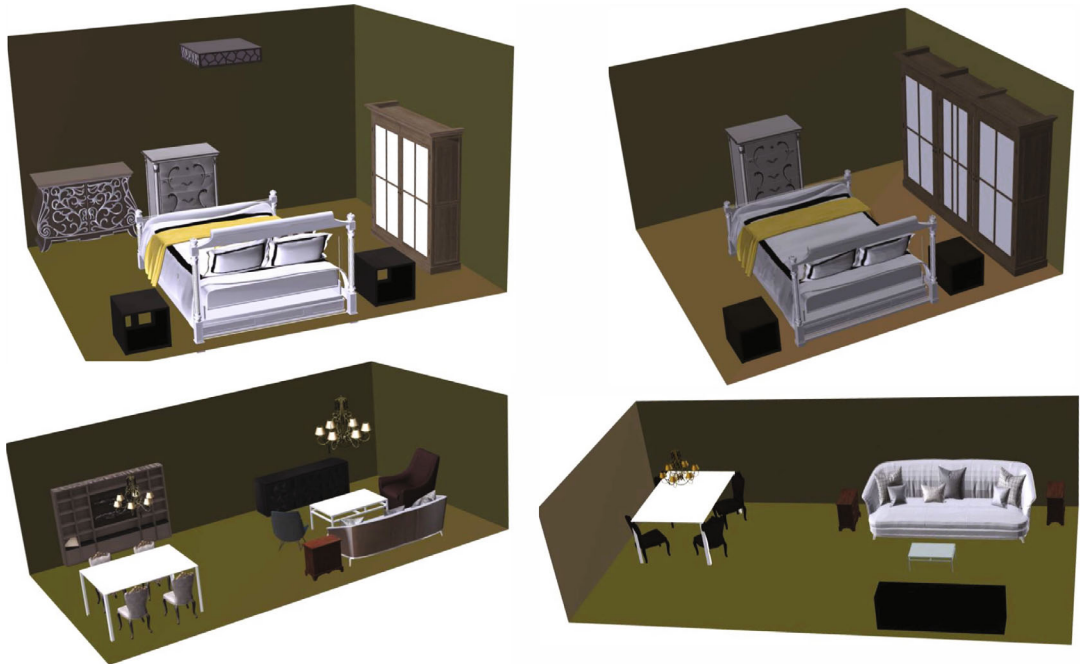

Figure 8: Nearest neighbor analysis of synthesis scenes. The first column showcases scenes generated by our method, while the second column presents corresponding nearest real scene from the training set.

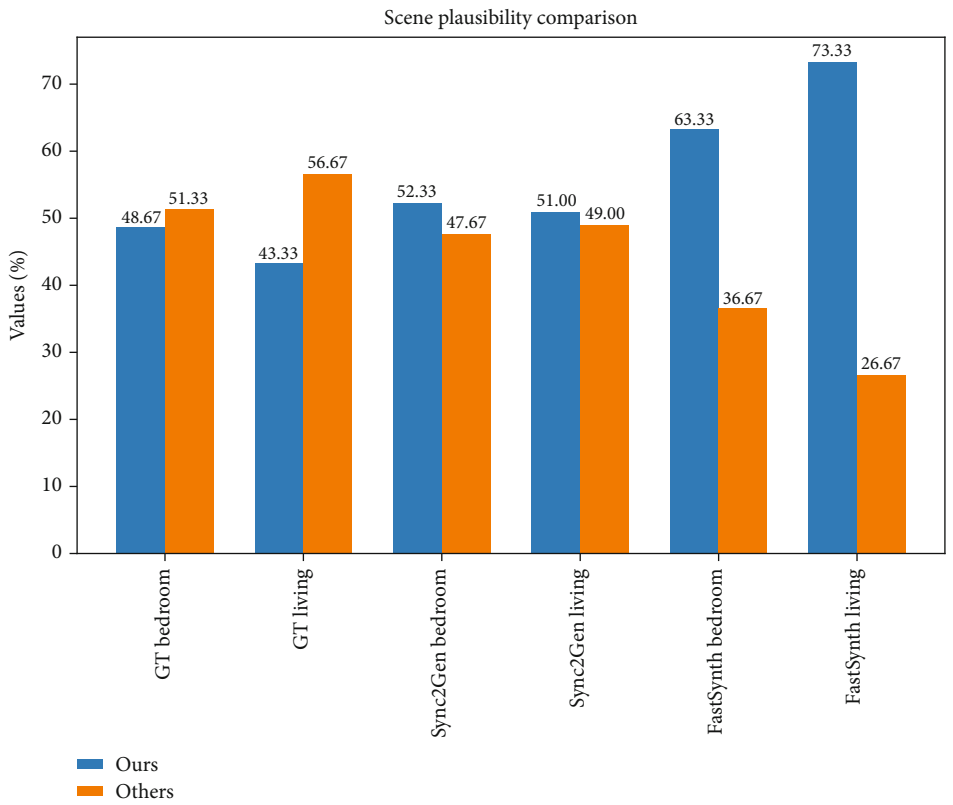

Figure 9: Perception research bar chart.

first column presents the scenes generated by our method, while the second column exhibits the scenes from the training set that bear the most resemblance.

For the bedroom scenes, in the first row of Figure 8, both our generated scene and its corresponding scene from the training set contain similar elements such as a bed, two nightstands,

a wardrobe, and a cabinet. However, our generated scene also includes a pendant lamp and an additional cabinet, and the wardrobe's size differs from its counterpart in the training scene.

In the case of living room scenes, our generated scene diverges more markedly from its most similar scene in the training set, featuring extra elements such as a single sofa, a chair, a pendant lamp, and a bookcase. These results showcase the discernible differences between our generated scenes and the real scenes in the training set, indicating that our model does not merely memorize the training data but effectively captures a range of inherent feature patterns within the scene data, thus demonstrating robust generalization capabilities. Notably, we determine the distance between scenes by calculating the Euclidean distance between their data matrices. This is achieved by computing the Frobenius norm of the difference between each generated scene's data matrix and those of the training scenes, thereby identifying the most similar scene.

4.3.4. Subjective Assessments. To comprehensively assess the realism of scenes produced by our method, we conducted a perceptual evaluation involving nonexpert users. We presented 30 participants with the same 60 pairs of scene images, asking them to choose the most visually appealing result from each pair. The findings are depicted in Figure 9.

It is evident that our method surpasses the two baseline methods, exhibiting a distinct edge particularly in generating bedroom scenes. Moreover, when juxtaposed with the ground truth, our method's virtually generated scenes hold their own against real 3D indoor scenes. These findings substantiate that our method is capable of producing visually engaging results.

5. Conclusions

For improving the efficiency, stability, and diversity of 3D indoor scene generation, we utilize matrix and graph representations to depict object attributes and the spatial relationships among objects within the scene, respectively. Leveraging the strengths of both VAE and GAN via the VAE-GAN module, we generate synthetic object attributes, and, using the U-Net module, we produce relative attributes of these synthetic objects. Ultimately, through the Bayesian scene optimization framework, we amalgamate scene prior, object attributes, and relative attributes to optimize and yield optimal object attributes. Our deep generative network is adept at comprehensively learning the adjacency, spatial, and embedding relationships of objects within indoor scenes, thereby producing highly reasonable 3D indoor scene layouts with extensive diversity.

Nonetheless, our present approach exhibits certain limitations, including its failure to take into account the styles of the objects within the indoor scenes. Our model is primarily geared towards the spatial location data of the objects in the scene, such as object position, orientation, and size, inadvertently neglecting the harmonizing styles of objects throughout the scene. In future research, we aim to incorporate the style information of the objects into our model to achieve consistently styled indoor scenes. Another drawback is that our model exclusively concentrates on generating regular scenes, i.e., rectangular rooms, without consideration for the generation of nonrectangular rooms. Although regular scenes predominate in daily life, irregular scenes remain a crucial component, and automating their layout generation would offer significant utility. Moving forward, we intend to delve into the layout generation of irregular scenes, aiming to enhance the applicability of our model.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This research is supported by the Teaching Reform Project of the Beijing Forestry University (grant number BJFU2020JYZD007].

References

[1] K. Wang, M. Savva, A. X. Chang, and D. Ritchie, "Deep convolutional priors for indoor scene synthesis," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–14, 2018.

[2] D. Ritchie, K. Wang, and Y.-a. Lin, "Fast and flexible indoor scene synthesis via deep convolutional generative models," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6182–6190, Long Beach, CA, USA, 2019.

[3] M. Li, A. G. Patil, X. Kai et al., "Grains: generative recursive autoencoders for indoor scenes," *ACM Transactions on Graphics*, vol. 38, no. 2, pp. 1–16, 2019.

[4] Z. Zhang, Z. Yang, C. Ma et al., "Deep generative modeling for scene synthesis via hybrid representations," *ACM Transactions on Graphics*, vol. 39, no. 2, pp. 1–21, 2020.

[5] H. Yang, Z. Zhang, S. Yan et al., "Scene synthesis via uncertainty-driven attribute synchronization," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5630–5640, Montreal, QC, Canada, 2021.

[6] E. R. Chan, C. Z. Lin, M. A. Chan et al., "Efficient geometry-aware 3D generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16123–16133, New Orleans, Louisiana, USA, 2022.

[7] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improved variational inference with inverse autoregressive flow," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[8] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, https://arxiv.org/abs/1312.6114.

[9] J. Adler and S. Lunz, "Banach wasserstein GAN," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[11] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[12] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," 2016, https://arxiv.org/abs/1609.03126.

[13] H. Fu, B. Cai, L. Gao et al., "3D-FRONT: 3D furnished rooms with layouts and semantics," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10933–10942, Montreal, QC, Canada, 2021.

[14] C. R. Qi, H. Su, M. NieBner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5648–5656, Las Vegas, NV, USA, 2016.

[15] S. Hang, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 945–953, Santiago, Chile, 2015.

[16] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Multi-view 3D models from single images with a convolutional network," in *European Conference on Computer Vision*, pp. 322–337, Springer, 2016.

[17] C. Häne, S. Tulsiani, and J. Malik, "Hierarchical surface prediction for 3D object reconstruction," in *2017 International Conference on 3D Vision (3DV)*, pp. 412–420, Qingdao, China, 2017.

[18] R. Klokov and V. Lempitsky, "Escape from cells: deep kd-networks for the recognition of 3d point cloud models," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 863–872, Venice, Italy, 2017.

[19] G. Riegler, A. O. Ulusoy, H. Bischof, and A. Geiger, "OctNetFusion: learning depth fusion from data," in *2017 International Conference on 3D Vision (3DV)*, pp. 57–66, Qingdao, China, 2017.

[20] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 652–660, Honolulu, HI, USA, 2017.

[21] J. Li, X. Kai, S. Chaudhuri, E. Yumer, H. Zhang, and L. Guibas, "Grass: generative recursive autoencoders for shape structures," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–14, 2017.

[22] S. Tulsiani, H. Su, L. J. Guibas, A. A. Efros, and J. Malik, "Learning shape abstractions by assembling volumetric primitives," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2635–2643, Honolulu, HI, USA, 2017.

[23] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," 2015, https://arxiv.org/abs/1506.05163.

[24] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on Riemannian manifolds," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pp. 37–45, Santiago, Chile, 2015.

[25] Z. Cao, Q. Huang, and R. Karthik, "3D object classification via spherical projections," in *2017 International Conference on 3D Vision (3DV)*, pp. 566–574, Qingdao, China, 2017.

[26] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling, "Spherical CNNs," 2018, https://arxiv.org/abs/1801.10130.

[27] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[28] C. Nash and C. K. I. Williams, "The shape variational autoencoder: a deep generative model of part-segmented 3D objects," *Computer Graphics Forum*, vol. 36, no. 5, pp. 1–12, 2017.

[29] K. Chen, Y.-K. Lai, W. Yu-Xin, R. Martin, and H. Shi-Min, "Automatic semantic modeling of indoor scenes from low-quality RGB-D data using contextual information," *ACM Transactions on Graphics*, vol. 33, no. 6, pp. 1–12, 2014.

[30] M. Sung, S. Hao, V. G. Kim, S. Chaudhuri, and L. Guibas, "ComplementMe," *ACM Transactions on Graphics*, vol. 36, no. 6, pp. 1–12, 2017.

[31] M. Fisher, D. Ritchie, M. Savva, T. Funkhouser, and P. Hanrahan, "Example-based synthesis of 3D object arrangements," *ACM Transactions on Graphics*, vol. 31, no. 6, pp. 1–11, 2012.

[32] Z. S. Kermani, Z. Liao, P. Tan, and H. Zhang, "Learning 3D scene synthesis from annotated RGB-D images," *Computer Graphics Forum*, vol. 35, no. 5, pp. 197–206, 2016.

[33] K. Wang, Y.-A. Lin, B. Weissmann, M. Savva, A. X. Chang, and D. Ritchie, "Planit: planning and instantiating indoor scenes with relation graph and spatial prior networks," *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 1–15, 2019.

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: a large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, Miami, FL, USA, 2009.

[35] A. X. Chang, T. Funkhouser, L. Guibas et al., "Shapenet: an information-rich 3D model repository," 2015, https://arxiv.org/abs/1512.03012.

[36] B. Allen, B. Curless, and Z. Popović, "The space of human body shapes," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 587–594, 2003.

[37] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "Scape: shape completion and animation of people," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 408–416, 2005.

[38] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH '99*, pp. 187–194, Los Angeles,USA, 1999.

[39] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *International Conference on Machine Learning*, pp. 1747–1756, New York City, NY, USA, 2016.

[40] A. B. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *International Conference on Machine Learning*, pp. 1558–1566, ew York City, NY, USA, 2016.

[41] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[42] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1746–1754, Honolulu, HI, USA, 2017.