JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 38, 1243-1263 (2022)
DOI: 10.6688/JISE.202211_38(6).0008

A Systematic Literature Review of Volumetric 3D Model Reconstruction Methodologies Using Generative Adversarial Networks

RILEY BYRD, KULIN DAMANI, HONGJIA CHE,
ANTHONY CALANDRA AND DAE-KYOO KIM+
Department of Computer Science and Engineering
Oakland University
Rochester, MI 48309, USA
E-mail: {rileybyrd; kldamani; chongjia; acalandra; kim2}@oakland.edu

3D modeling is increasingly pervasive in many industries to produce a 3D digital representation of any object. Nonetheless, traditional 3D modeling remains a laborious and expensive undertaking, requiring a high degree of expertise and patience to create realistic models. GANs have shown great promise in the application of 3D object reconstruction and there has been a vast amount of research being conducted on this topic in recent years. However, given the many potential fields of application for GANs, little work has been produced on the study of current state-of-the-art methods and what kind of future uses they may have. In this paper, we present a systematic literature review of the current unsupervised and weakly-supervised methods on volumetric 3D object reconstruction utilizing GANs with a voxel representation. The review aims at offering insights into future works based on the constraints and potentials of the studied works.

Keywords: generative adversarial networks, literature review, object reconstruction, survey, 3D model, voxel

1. INTRODUCTION

In the field of 3D computer graphics, 3D modeling is the process of developing a digital representation of some object. This is carried out by creating and manipulating points within a simulated 3D coordinate space. This process is used by many industries for various purposes including anything from computer-aided design for the purpose of manufacturing items at an incredible level of precision to the development of art assets for special effects in movies and video games.

Despite their widespread use and continued growth in popularity, high quality 3D models remain expensive and difficult to produce. Experts are always in high demand and the process of creating 3D models continues to remain time consuming despite recent advances in tools and technology. Even the most popular contemporary programs like CATIA, SolidWorks, and Inventor, fail to meaningfully reduce the time and effort requirements industry professionals face [1]. With Generative Adversarial Networks (GANs) [2],

Received November 1, 2021; revised December 12, 2021; accepted January 26, 2022.
Communicated by Wei-Ta Chu.
+ Corresponding author.

however, new possibilities have been presented in creating high quality 3D models in record time with minimal or even no human supervision.

GANs have been able to reconstruct convincingly real and natural-looking synthetic data by learning about the features of the given data. This is performed by exploring what is called the latent space which is the area where vector arithmetic is done on said data of an image or object being constructed [3]. Using 2D image data, GANs can learn how to generate 3D models. However, there is a problem when these 3D shapes are rendered only using 2D images. In order to create an exact replica from input data, a topology representing the replica should be constructed. A practical way of constructing a topology with ease of use is using voxels. A voxel is an abstract 3D unit value with pre-defined characteristics which can be used to represent a topology in a filled space [4]. By using voxel-based representations in conjunction with GANs, a topological representation of a reconstructed model can be created [5].

Although GANs are a relatively new class of machine learning frameworks, being introduced only recently in 2014 [2], their growth in terms of related work produced in academia has been nothing short of explosive. While this growth has opened up a vast number of potential application fields, a field of popular interest has been found in applying GANs to the generation and reconstruction of 3D models.

While there is much existing work on the topic of GANs and their use in 3D model reconstruction, given the extensive array of applications for GANs, there exists little work on the topic of comparative analysis of GANs being applied for volumetric 3D reconstruction with the use of voxel representation within the latent space. In this paper, we present a systematic literature review of the current state-of-the-art methods on this topic in what is, according to our findings, the first of its kind.

The review follows the guidelines provided by Kitchenham *et al*. [6] by first establishing a selection procedure and then developing the research questions to be answered within this work. Following that, primary works that fit the selection criteria are identified and reviewed. These works are analyzed with respect to the parameters we established, so that we may attempt to answer the presented research questions.

The structure of the remainder of the paper is as follows. Section 2 gives an overview on the basic architecture of GANs, their framework methodology, and the distinction of how GANs have been applied to produce both 2D and 3D outputs. Section 3 summarizes and discusses literature reviews relevant to the topic. Section 4 details the exact approach to this survey along with findings. Section 5 presents the answers to the research questions posed with regard to the data collected. Finally, Section 6 gives a conclusion to this work.

2. BACKGROUND

In this section, we give an overview of GANs and its use in the construction of 2D and 3D objects. A GAN is a type of machine learning frameworks for learning to generate new data with the same data distribution as the training set. Figure 1 shows the general architecture of GANs. A GAN is generally composed of two primary components – the generator and the discriminator, both of which are separate neural networks and exist as the competitors of the adversarial process. The generator synthesizes new data and introduces the generative element to the GAN. The data is generated based on the

learning from the feedback of the discriminator as well as from the real samples provided in the training data. The objective of the generator is to deceive the discriminator. This is achieved by synthesizing an object that is indistinguishable from the real samples given during training. Likewise, the discriminator's objective is to judge the given data from the generator to determine if it is real or synthesized [2, 7]. These two components compete in a zero-sum game in which they each take turns performing their tasks and are then evaluated with a winner chosen. This learning is facilitated by an architecture where the generator outputs synthesized data and real examples are fed into the discriminator as input. Ideally, this process will begin with the discriminator having a high success rate and deteriorating as the generator becomes increasingly accurate. The end result is an exceptional generator model which can be used for some assigned purpose. The discriminator is also trained to the same capacity. Specifically, it utilizes back-propagation from its loss function, updating itself when a misclassification occurs. While both can be used for great effect elsewhere, the generator is what is commonly sought after [2, 7].

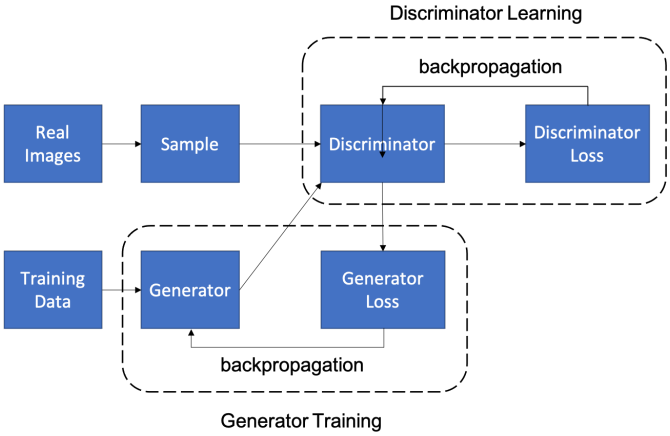


Fig. 1. General GAN architecture.

In recent years, GANs have undergone extensive research, especially in the field of image generation. The application of GANs has been extended to 3D modeling, naturally benefiting from the extensive 2D research that has been established. Zhu *et al*. [8] designed the operation to produce realistic results for the image editing. First, they used GANs to approximate the manifold and applied a hybrid method of feed-forward and optimization to project the input image to find the closest latent vector. Then, they used constraints to manipulate the latent vector to match the user's intent and applied gradient descent to stay close to the input in the manifold. Finally, the version is converted back to the original picture by applying the traditional optical flow method. Inspired by Zhu *et al*.'s work, Liu *et al*. [9] proposed a revised architecture of 3D-GAN [10] to maintain stability during training. The architecture projects original or modified 3D objects into the learned manifold by training three neural networks – a generator, a discriminator, and a projection operator. The generator accepts as input a latent vector representing the user model and the discriminator learns the latent space of manifolds. The generator inputs the 3D model to the projection operator which maps the input 3D model to its latent vector

so that it can be used in the generator. The 3D model is directly input instead of training additional networks. The projection operator adjusts the importance of the rationality of the generative model and its similarity with the original input. In their implementation, they used a feed-forward network to optimize the similarity (which is used as an initial guess) using gradient descent to find a local minimum on the entire target loss. In this case, calculation time is important due to its interactivity.

3. RELATED WORK

While there exist many literature reviews regarding GANs, they mostly focus on quantitative analysis of GAN applications as a whole. Aggarwal *et al*. [11] surveyed recent works on GAN models for the purpose of identifying what practical, real-world, application fields currently exist for GANs under present academia. Identified fields include medicine, image processing, face detection, physics, and astronomy. Of all the application fields, medicine contained the largest amount of studies that defined distinct applications within the respective field. Majority of these key studies are related to the identification and classification of medical images and their contents. Examples include sharpening and de-noising magnetic resonance images and classifying cytological images. They also discuss some work on 3D image generation. The key studies [12–16] identified covered a variety of approaches to the topic. One study [12] proposed a GAN that encodes unclear, unlabelled data to be processed and decodes the encoded data into a more precise representation of the object as a point cloud. Another study [15] focused on training a GAN to learn how clothing deforms in accordance to human motion. The work concludes with a brief section on the limitations of GANs. However, the presented limitations are more correctly described as moral and ethical concerns regarding the technology rather than practical limitations such as computational requirements.

Another survey by Shamsolmoali *et al*. [17] performs a comprehensive review of adversarial models that are used for image synthesis as well as the various categories of methods that accompany the models. A subset of the methods discusses applications to 3D shape reconstruction, specifically the generation of data to fill in where there are missing components. Multiple architectures that contain varying implementations to the presented problem are reviewed. One model [10] utilizes volumetric convolutional networks in conjunction with GANs to generate a 3D object from a probabilistic space. The model is relatively simple in that it maps the lower-dimensional probabilistic space from the 2D image input and transforms it into a 3D probabilistic space which results in the generated model. The survey notes that while the model is indeed architecturally simplistic and therefore likely to perform well, the evaluation parameters and data are limited, requiring additional work to confirm the claim. Another model [18] adopts a Wasserstein normalization that incorporates gradient penalization during training. While this model provides generation of realistic images, the architecture itself can be used for 3D shape reconstruction from a 2D image with subsequent shape completion functions. The survey concludes with discussions on the difficulties of image synthesis tasks as well as the need for further studies to be made to better understand the practical limitations of these models in terms of accuracy, training time, and testing time with regards to the field of computer vision.

The most similar work to ours is by Han *et al*. [19] where a comprehensive review is conducted on image-based 3D object reconstruction using deep learning techniques. Unlike our work, however, voxel representation is covered in addition to surface and point-based representations. The work further distinguishes itself from ours by including architectures that do not utilize adversarial learning methods, and therefore, are not limited to GANs. While some analysis parameters are shared with our work, such as training strategies and evaluation methods, emphasis is placed on the analysis of the different output representations. Performance of selected algorithms is also a point of focus, both in terms of computational efficiency and in output accuracy. The work concludes with suggestions on future research paths to take in order to refine existing capabilities as well as to alleviate potential issues such as the limited quantity and quality of publicly available training datasets. Han *et al*.'s individual suggestions contribute toward their final suggestion which posits that semantic parsing of a complete 3D scene from one or more images is the ultimate goal of the technology.

The existing literature reviews are plentiful and offer excellent insight into the current state of GANs in academic literature. Our work differs from the existing reviews in that our work focuses the application of GANs to 3D object generation and reconstruction in voxel representation. Other representations, such as meshes, are not considered in order to provide a more concise analysis. Different representations may require different implementations which could interfere with our findings.

4. SYSTEMATIC LITERATURE REVIEW

In this section, we present a systematic literature review of the current state of volumetric 3D object reconstruction in voxel representation by using GANs. The review is structured in a four step process according to the guidelines provided by Kitchenham *et al*. [6]. First, the review method to apply is formulated. Second, a review is conducted based on the review method. Third, the results of the review are reported in a quantitative manner. Finally, the findings from those results are discussed. The review method consists of a set of research questions, academic database searching, identification and evaluation of primary works found, and parameterized data extraction from the primary works. By conducting our systematic literature review, we hope to answer the following research questions.

1. What commonalities can be extracted from analyzing the current methods of 3D object generation using GANs?
2. With these commonalities in mind, what inferences can be made about the state of the application of GANs?
3. What are limitations on a general scale?
4. What recommendations can be made towards future work?

4.1 Selection Process

Primary works included within this review were obtained by searching Google Scholar which indexes many major computer science-related publications. Preliminary

filtering of search results was achieved by utilizing Google's advanced search operators. By identifying key terms related to our topic and including similar terms to increase the accuracy of the search results, we arrived at our final search string: "("3d object" OR "3d shape") AND ("reconstruction") AND ("voxel") AND ("latent space") AND ("GANs" OR "generative adversarial network") AND ("unsupervised" OR "weakly supervised") -face -hands -"literature review" -survey -"point cloud"". Quotations denote exact phrases that must appear at least once, while hyphens denote excluded terms. The AND and OR operators function as they do in logic operations, requiring precise combinations and allowing set alternative combinations, respectfully.

As discussed in Section 3, there is a wealth of existing work regarding GANs. However, many works focus on using GANs for purposes outside of the scope of our systematic literature review. Therefore, we put forward a set of criteria that must be met by any work to be included within this review. This criteria is as follows.

- The work must be written in English.
- The work must be available in full, either freely or within the resources accessible to our institution.
- The work must be published between 2018 and 2021 to consider up-to-date works.
- The work must have a primary focus on the application of GANs for use in 3D object reconstruction.
- The work must use voxel representation.

Table 1 shows the list of identified works. While we made sure to include all relevant works given our constraints, we acknowledge that some may have been missed. However, no works were intentionally left out and we believe our current pool of selected works still offers valid and valuable insight into our topic.

Table 1. Selected works.

Year	Study	Venue
2020	Inverse Graphics (IG) GAN [5]	arXiv
2020	3DMaterialGAN [20]	arXiv
2020	Deep Convolutional Refined Auto-Encoding Alpha GAN [21]	IEEE TMRB
2019	MP-GAN [22]	arXiv
2019	3D-VAE-Stack-SNGAN [23]	CCP&E
2019	3DMaskGAN [24]	ICBESCC
2019	Chen *et al*. [25]	ICIP
2019	Chen *et al*. [26]	MMM
2018	Yang *et al*. [27]	ECCV
2018	ORGAN [28]	CGI

4.2 Structural Analysis

We analyze the identified works in terms of generator, discriminator, GAN type, training strategy, loss function, other noteworthy components, evaluation method, and limitations. Table 2 shows the summary of the analysis.

Table 2. Analysis parameters.

Ref.	Generator	Discriminator	GAN Type	Training Strategy	Loss Function	Other Components	Evaluation Methods	Limitations
[5]	3D CNN	Novel	IG-GAN	ShapeNet	Discriminator network	Arbitrary renderer, novel proxy neural renderer, classifier	Inception network	Potential inconsistency of alterations
[20]	Novel	Novel	3DMaterialGAN	ShapeNet & Titanium	Unspecified	N/A	Comparing against existing models	Difficult to evaluate the system
[21]	3D CNN	Novel	Deep Convolutional Refined Auto-Encoding Alpha GAN	Alzheimer's Disease Neuroimaging Initiative	Unspecified	Code discriminator, refiner, VAE	Jaccard Index	Impractically low resolution and difficult to handle blood vessels
[22]	3D CNN	Multiple*	MP-GAN	ShapeNet	Discriminator network	Projector, classifier	FID score	Cannot model concavities
[23]	3D CNN	Multiple*	3D-Stack-SNGAN & 3D-VAE-Stack-SNGAN	ModelNet10	Discriminator network	VAE	Inception network	Unspecified
[24]	3D CNN	Novel	3DMaskGAN	ShapeNet	2D masking	Projector	Unspecified	Unspecified
[25]	SSCNet	Novel	Unnamed	Depth images	Discriminator network	N/A	Jaccard Index	Inferior performance to state-of-the-art
[26]	3D CNN	Novel	Unnamed	ModelNet10	Wasserstein loss	Classifier, VAE	Jaccard Index	Unspecified
[27]	3D CNN	3D CNN	Unnamed	ShapeNetCore	Unspecified	Encoder, projector	Jaccard Index	Working with arbitrary camera locations
[28]	3D CNN	3D CNN	ORGAN	ModelNet10 & novel object class	Novel	Encoder	Comparing object fragment sizes	Instability in training

CNN: Convolutional Neural Network; FID: Fréchet Inception Distance; SSCNet: Semantic Scene Completion Network; VAE: Variational Auto-Encoder
*: Stack structure

4.2.1 Generator

The architecture of the generators in the studied works is found consistent across the studied works. 50% (*i.e.*, [23, 24, 26–28])of the generators output their objects at a resolution of $32\times32\times32$ voxels, followed by 40% (*i.e.*, [5, 20–22]) at $64\times64\times64$. This is unsurprising, as memory requirements are cited as the limiting factor to outputting at a higher resolution [21]. The remaining generator from Chen *et al*.'s work [25] outputs data at an unusual resolution of $60\times36\times60$ voxels. This is because the generator is simply SSCNet, an end-to-end 3D convolutional network by Song *et al*. [29] that requires a $240\times140\times240$ depth image as its input, rather than the usual normalized latent vector.

While 80% (*i.e.*, [5, 21–24, 26–28]) of the generators (including those outputting a lower resolution) followed the standard neural network architecture as proposed by Wu *et al*. [10], there is one notable exception. 3DMaterialGAN [20] describes a novel generator that begins with an eight-layer mapping network that transforms the latent vector into an intermediate latent space. This intermediate form is then transformed again and passed through a five-block synthesis network where each block is similar to StyleGAN by Kerras *et al*. [30].

4.2.2 Discriminator

The discriminator in a GAN is simply a classifier. It tries to distinguish real data from the data created by the generator. It can use any network architecture appropriate to the type of data it is classifying. The discriminators that were used in the review vary. 20% (*i.e.*, [27,28]) of the studied works used a unified model architecture which includes an encoder, generator and discriminator. Another 20% (*i.e.*, [22,23]) used multiple

discriminators in a stacked structure which can be considered as a coarse-to-fine or low-to-high–resolution mechanism. The remaining 60% [5, 20, 21, 24–26] of the works used novel discriminator models which were designed with respect to the project requirements.

Lunz *et al*. [5] proposed a novel loss function named *discriminator output matching* to train the proxy neural renderer. This novel loss function is required in order to account for the fact that the off-the-shelf renderer is non-differentiable, where as the generator is not. Employing the loss function forces the proxy neural renderer to generate images that smoothly interpolate the discrete and continuous forms of rasterization and voxel grids, respectively for the discriminators use. Without this novel loss function, the proxy neural renderer would be producing arbitrary outputs for the generated voxel grids.

In Chen *et al*.'s work [25], the discriminator network takes, at random, either the generated 3D volume or the ground truth volume as input and classifies the data as real or fake. The parameters of each layer are shown as *the number of filters, kernel size, stride* in the case of convolutions and as *the number of output channels* in the case of fully connected layers. The aim of the discriminator network is to distinguish a generated 3D volume from a ground truth volume. To this end, they transform a ground truth sample of the training data to a volume of the same size using one-hot encoding.

A Code Discriminator (CD) and refiner were used for the network structure by Segato *et al*. [21]. The CD network consists of three fully-connected layers. LeakyReLU and BatchNorm layers are placed between each pair of the three layers. The CD is trained to distinguish between latent vectors coming from the variational autoencoder and the random ones given as input to the generator. This adversarial process makes the probability distributions of the two latent vectors match, reducing the image blurriness that is characteristic to variational autoencoder outputs. The architecture of the refiner consists of four ResNet blocks. In traditional neural networks, each layer feeds into the next. In a network with residual blocks, each layer feeds into the next and also directly into the layers roughly 2–3 hops away. The presence of skip connections reduces the vanishing gradient problem. It smooths the shapes of the image and allows the generation of more realistic outputs.

Li *et al*. [22] trained the generator network with cues from multiple discriminators in parallel. Each discriminator operates on the subset of the training data corresponding to a particular viewpoint and is trained from independently drawn samples from a silhouette image. 3D-VAE-Stack-SNGAN [23] uses multiple generators and discriminators to enhance the ability of the model for learning complex distributions. This stacked structure can be considered as a coarse-to-fine or low-to-high–resolution mechanism. The spectral normalization technology is employed to control the Lipschitz constant of the discriminators by literally constraining the spectral norm of each layer to get a more stable training process, allowing the proposed model to generate realistic and high-quality 3D objects. The discriminator model weights are updated five times in each training iteration, while the generator model weights are updated only once in order to minimize their respective losses.

Wan *et al*.'s work [24] used a 2D convolution operation in lieu of the 3D convolutional neural networks in the discriminator. This reduces the number of iterations, while increasing the speed of training. An unified model architecture was proposed by Yang *et al*. [27] with three main components – encoder, generator, and discriminator. The encoder takes an image, a silhouette mask, as its input and produces a latent representation

of shape. The generator takes the latent representation as input and produces a voxel grid. The discriminator tries to distinguish between rendered views of the voxel output by the generator and views of the real objects. In Chen *et al*.'s study [26], the discriminator is a binary classifier that tries to differentiate real images from generated images. Different from the traditional GAN formulation, the discriminator accepts input-label pairs produced by the classifier and the generator. They sample ground truth input-label pairs (x, y) from the data and label them as real pairs. For fake pairs, they pair real images x with predicted labels coming from the classifier forming an input-label pair (x, $C(x)$) where C is the classifier. 3DMaterialGAN [20] is a simple GAN that consists of a generator and a discriminator. The generator tries to synthesize the samples that look like the training data, while the discriminator tries to determine whether a given sample is a real sample originated from the ground truth data or from the generator. The discriminator D outputs a confidence value $D(x)$ of whether input x is real or synthetic.

4.2.3 GAN types

Among the studied work, only 30% (*i.e*., [25–27]) of the works used existing GAN types to achieve their goals, while the other 70% (*i.e*., [5, 20–24, 28]) proposed novel methods by using the existing GAN types.

The work by Lunz *et al*. [5] aimed at training a generative model for 3D shapes such that rendering these shapes with an off-the-shelf renderer generates the images that match the distribution of a 2D training image dataset. Through trying a number of different methods and addressing the issues during the generation process, a novel method named Inverse Graphics GAN was proposed as the neural renderer during backpropagation"inverts" the off-the-shelf renderer providing useful gradients for the 3D generative model training.

The work by Chen *et al*. [25] addresses 3D semantic scene completion by predicting the semantic labels and occupancy of voxels in the 3D geometry of the objects in the scene of a given single depth image. Their results show that using conditional GANs outperforms the vanilla GAN setup when evaluating their architecture designs on several datasets. Based on the experiments, GANs are demonstrated to be able to outperform the performance of a baseline 3D convolutional neural network in the case of clean annotations, but perform poorly on improperly-aligned annotations.

Segato *et al*.'s work [21] proposed Deep Convolutional Refined Auto-Encoding Alpha GAN, an innovative type of GAN that is able to successfully generate 3D brain magnetic resonance imaging data from random vectors by learning the data distribution. This is done by combining a variational autoencoder GAN with a code discriminator to solve the common mode collapse problem and reduce the image blurriness. A refiner is inserted in series with the generator network in order to smooth the shapes of the images and generate more realistic samples.

Li *et al*. [22] presented a novel GAN type called Multi-Projection GAN (MP-GAN) that can generate 3D shapes with only unoccluded silhouette annotations from a categorized object. MP-GAN is unique in that the output generator is trained without ever having direct access to the objects data. Instead, it is given a set of 2D projections from multiple discriminators and must assess if each projection belongs to a certain object category. Each discriminator is able to be trained independently and can therefore have samples with differing objects or projection parameters from the other discriminators. This allows

the output generator to be trained on different objects and projection parameters without needing to have explicitly defined correspondences.

3D-Stack-SNGAN [23] generates high-quality 3D objects and outperforms the compared state-of-the-art method. Multiple generators and discriminators are used to build a stacked structure to make the model learn complex distribution more effectively, and use spectral normalization on the discriminators to increase the stability of the training process. 3D-Stack-SNGAN is combined with variational autoencoders as 3D-VAE-Stack-SNGAN which is used for 3D object recovery tasks. The experiments demonstrated that the system can generate and recover realistic and high-quality 3D objects.

Hermoza and Sipiran [28] presented ORGAN, a GAN focused on the reconstruction and completion of damaged archaeological objects. ORGAN is trained by taking complete objects, randomly sampling occupied voxels in the grid space to simulate fractures, and then training the shape completion network on the samples. Citing instability during training as a common issue for GANs as a whole, a novel loss function is also presented that combines the stability and quality improvements of Improved Wasserstein GAN (IWGAN) [31] with the class label-based training of a Conditional GAN (CGAN) [32]. The result is a model that can recreate the majority of an objects structure with minimal errors, even with the fragments that account for less than half of the total objects volume.

Inspired by IWGAN [31], 3DMaskGAN proposed by Wan *et al*. [24] reconstructs the full 3D shapes of objects from an arbitrary image. By exploiting the generalization capabilities of the generation network and masked discriminator, 3DMaskGAN can predict decent 3D shapes with less iteratively trained models and still outperform other networks under the same conditions. The training results show that the training process achieves minimal loss with only 100 iterations and significantly shortens the overall training time. 3DMaskGAN only trained 100 epochs to lower the discriminator loss to 0.01, while IWGAN was required to train 1000 epochs.

Yang *et al*. [27] proposed a unified and end-to-end model that uses both images labeled with a camera pose and unlabeled images as supervision for a single view 3D reconstruction, and evaluated different training strategies with limited annotations. PrGAN [33] and DRAGAN [34] are used in training and implementation to achieve adversarial loss and improve training stability. The experiment results showed that one can train a single-view reconstruction model with minimal pose annotations when leveraging unlabeled data.

The semi-supervised method proposed in Chen *et al*.'s work [26] can recover the complete shape of a broken or otherwise an incomplete 3D object model, and built a hybrid of a 3D variational autoencoder and a GAN to recover the complete voxelized 3D object. A separate classifier was incorporated in the GAN framework which helps stabilize the training of the GAN as well as guide the shape completion process to follow the object class labels. VAEGAN [35], CVAEGAN [36], and 3DIWGAN [18] are used in the experiment as contrasts to their method. The experiments showed that the model produces 3D object reconstructions with high-similarity to the ground truths and outperforms several baselines in both quantitative and qualitative evaluations.

With a focus on applying GANs to materials science problems, Jangid *et al*. [20] presented 3DMaterialGAN. Built off of StyleGAN by Karras *et al*. [30], 3DMaterialGAN specializes in identifying and generating 3D objects of crystalline material microstructures. The formation of these microstructures are influenced by a variety of factors and

therefore cannot be evaluated reliably by more simple means such as direct object comparison. The architecture begins with a mapping network that takes in a latent vector. An intermediate latent space is outputted and passed to a five-block synthesis network before being outputted.

4.2.4 Training strategy

Training strategy is a noteworthy element of any artificial intelligence based solution as it can often be leveraged as a tool to squeeze maximum efficiency, precision, and accuracy out of an application. Datasets are a critical element to any training strategy. Table 3 shows the datasets used in the studied works. In the table, it is observed that 50% (*i.e.*, [5, 20, 22, 24, 27]) of the studied works used ShapeNet [37] or a variation of it and 30% (*i.e.*, [23,26,28]) used ModelNet [38]. However, with this similarity, there is variance in the number of object categories selected ranging from three up to ten categories. This may highlight the fact that despite strong datasets and volume of data often being a rare and limiting factor in GAN research, there can be high variance in how the datasets are utilized to create unique results and environments from these datasets.

Table 3. Training datasets.

Study	Datasets
IG-GAN [5]	ShapeNet with categories of chairs, couches, bathtubs
3DMaterialGAN [20]	ShapeNet with categories of car, chair, plane, guitar, sofa, rifle plus Titanium
Deep Convolutional Refined Auto-Encoding Alpha GAN [21]	Alzheimer's Disease Neuroimaging Initiative dataset (ADNI)
MP-GAN [22]	ShapeNet with category of chairs
3D-VAE-Stack-SNGAN [23]	ModelNet10
3DMaskGAN [24]	ShapeNet with chair, sofa, table, car and boat categories
Chen *et al*. [25]	Depth Images
Chen *et al*. [26]	ModelNet10
Yang *et al*. [27]	ShapeNetCore with categories of airplanes, cars, chairs, displays, phones, speakers, tables, benches, vessels, and cabinets.
ORGAN [28]	ModelNet10 with additional "archaeological looking" object class

Another critical element of a training strategy is supervision level. GAN networks are often unsupervised or in some cases semi-supervised. 70% (*i.e.*, [1,5,20,21,23,25,28]) of the studied works are unsupervised and 30% (*i.e.*, [22, 26, 27]) are semi-supervised. While unsupervised learning is dominant, semi-supervision is becoming prevalent. In unsupervised learning, a large amount of unlabelled datasets are critical to training GAN networks. The unsupervised works studied in this survey make use of ModelNet as their training dataset for its effectiveness in pairing with unsupervised learning. ModelNet also provides a large amount of data points to create the diversity and quantity needed for unsupervised learning. Some works [22, 26, 27] demonstrate the use of unconventional semi-supervised learning models to address the challenge of object completion and reconstruction. In Chen *et al*.'s work [26], semi supervision exists in that a third neural network is included in the game to act as the object classifier giving the other networks

a general classification of training data, which improves the learning rate of the network and increasing the network's stability. In MP-GAN [22], 2D silhouette annotations on 3D training objects are provided, allowing for a partial labeling of training data. This application of partial labeling considerably reduced the cost of learning for the 3D shape generator for new categories. From these works, it is observed that the application of weak supervision provides favorable improvements in the system performance, while increasing complexity.

4.2.5 Loss function

The loss function measures how close an estimated or random value is from a true or real value. The measured difference can be referred to as a "cost" associated with what is being tested. Three different types of methods are used for the loss function in the studied works – discriminator networks, 2D masking, and Wasserstein loss. 40% (*i.e.*, [5, 22, 23, 25]) of the studied works used discriminator networks, 10% (*i.e.*, [24]) used 2D masking, 20% (*i.e.*, [22,26]) used Wasserstein loss, while the remaining 30% did not mention loss functions used.

The usage of a discriminator network The most popular method is using a discriminator network which is used by 40% (*i.e.*, [5,22,23,25]) of the surveyed papers. Each work employs their discriminator network differently, according to the needs of the model. For 3D-Stack-SNGAN [23], the stacked structure is taken advantage of by employing multiple discriminators to evaluate the generators output at the three largest resolutions in the stack: $32\times32\times32$, $16\times16\times16$, and $8\times8\times8$. By evaluating that the generator systems work at multiple levels in tandem, both the quality and diversity of generated objects are increased, according to evaluations conducted against similar contemporary methods.

The work by Lunz *et al*. [5] also discusses the backpropagating of the gradient from the GAN discriminator through the neural renderer to the 3D generative model, allowing training using gradient descent. This is to match the rendering output given a 3D input from the off-the-shelf renderer. To address the non-differentiability issues, they introduce a proxy neural renderer for the rendering of continuous voxel representation. This allows for backpropagation in the generator during training, minimizing the loss error of rendering on discrete voxels.

Chen *et al*. [25] used two types of parameters for the hybrid loss function. One is the discriminator network as mentioned previously. Another is a multi-class cross-entropy loss which is used for the generator to predict class label at each voxel location independently. They denote the class probability map over the classes C for the volume $\mathrm{H}\times\mathrm{W}\times\mathrm{D}$ where H, W, and D are the height, width and depth of the 3D volume, respectively. This probability is produced by the generator network. Then, the loss is minimized according to the parameters of the generator network (*i.e.*, multi-class cross-entropy), while maximizing it with respect to the parameters of the discriminator network.

MP-GAN [22] uses a similar approach of minimizing or optimizing the parameters of the generator network. They take an approximation of 3D object distribution at an i-th scale and test it on three different generators to produce three different scales. This assists in returning to a small-to-large-scale structure to help the model learn simple to complex distribution. They also use the discriminator to minimize the loss function. By normaliz-

ing weight matrices, they can normalize each weight matrix of their discriminators, which minimizes the loss function.

2D masking from different viewpoints 10% (*i.e.*, [24]) of the methods found in the survey was masking of a 2D output from different mask viewpoints using binary cross-entropy. 3DMaskGAN [24] describes how the generation of 3D volume shapes and 2D masks can be completed at the same time. As this happens, the encoder and generator are penalized on the reconstruction error of samples. They select the reconstruction error on masks as the generators loss to improve unsupervised conditions. The generator loss takes into account the index of output 2D masks where different viewpoints of a target mask sample and a generated mask sample are considered. They also use the means and variances produced by the encoder and the discriminator and a confidence value of whether the mask is real or synthetic to compute the loss of the encoder and generator. As a classification of loss, they train the mask to discriminate real from generated masks using binary cross-entropy.

Wasserstein loss with a gradient coefficient 20% (*i.e.*, [26, 28]) of the studied works use Wasserstein loss with a gradient as their loss function. This involves the use of a linear activation function in the output layer of the discriminator. The Wasserstein loss function then trains the generator and discriminator to differentiate scores for real and generated images. They make use of ReLU for their activation function with no batch normalization except for the last fully connected layer which outputs a single value with no activation function. This then aids the output of the network to be part of the training data manifold.

4.2.6 Other components

This section discusses other noteworthy modules than the generator or discriminator within the network architecture of the studied works. The most common components are encoders, being present in 50% (*i.e.*, [21, 23, 26–28]) of the studied works. Of this percentage, 60% (*i.e.*, [21, 23, 26]) of the encoders are variational autoencoders, while the remaining 40% (*i.e.*, [27, 28]) are standard encoders. The prevalence of encoders is expected as they are used to transform data into a latent vector which is the input format for standard generator models. The next most commonly appearing components are projectors, occurring in 30% (*i.e.*, [22, 24, 27]) of works reviewed. In all the three works, the projector was positioned between the generator and the discriminator. This is because the projector is used to create silhouette masks which are 2D images of the generated object at a certain viewpoint from within a 3D space. Silhouette masks are one of multiple data formats used to train the discriminator. 30% (*i.e.*, [5, 22, 26]) of the studied works include classifiers positioned between the generator and discriminator in order to feed the discriminator additional information. IG-GAN [5] and the work by Chen *et al*. [26] use a classifier to predict object class labels to be used as conditional information by the discriminator. MP-GAN [22] uses a classifier to predict the viewpoint of a given silhouette mask. A work-specific component is also used in 10% (*i.e.*, [21]) of the studied works. Segato *et al*. [21] used a refiner network as an additional component to synthesize the images that have been smoothed to be closer to real data. The refiner is

trained separately, loading the weights of previously trained components. The remaining 30% did not contain any components classifiable to this subsection.

4.3 Evaluation Methods

Different methods are used in the studied works to evaluate their test results and overall approach performance . The methods can be categorized into Jaccard Index [21, 25–27] (40%), Inception network [5] (10%), FID score [22] (10%), fragment sizes [28] (10%), and checking against existing models [20, 23] (20%). The remaining 10% do not mention an evaluation method.

Jaccard Index The majority of the studied works (*i.e.*, [21, 25–27]) use Jaccard Index, also known as Intersection-over-Union (IoU), as their evaluation method. The IoU is measured between the ground truth voxel grid and the predicted one averaged over all the objects. Computing the IoU requires a threshold of the probability output of the voxels from the generator. Chen *et al*.'s work [25] uses the evaluation method by Liu *et al*. [39] on the tested dataset for comparison and to compute an average precision. The work of Yang *et al*. [27] uses previous methods suggested by Tulsiani *et al*. [40] to sweep thresholds and report the maximum average IoU. The number of samples for test per category varies among the studied works.

The work of Chen *et al*. [26] showcases their approach of improving an already existing training model by implementing the Tensorflow deep learning library and Tensorlayer. They have experimented with parameters of their model such as batch size, epochs, and the beta value. They also use an Adam optimizer to further improve their model. The evaluation is done using the Hausdorff Distance(HD) where the lower HD value the better the result. From the observations of training their model using IOU and HD, the model outperformed all baseline models.

Segato *et al*. [21] uses Jaccard Index (*i.e.*, IoU) to evaluate the similarity of two sets as a ratio of the number of common elements of the sets to the total number of elements of the sets. Then, the real and generated samples are compared with IoU scores at a voxel level where the higher IoU score the closer the distributions of the real and generated samples. Their results are shown quantitatively, but are evaluated by comparing an image from the training set and a generated image.

Inception Network Lunz *et al*. [5] use an Inception network [41] which renders a 3D model to a 2D image and computes an Inception using an Inception network trained to classify ShapeNet images made by their renderer. Lunz *et al*. [5] trained their Inception network rendered images in ShapNet and compared them against three other models – Visual Hull [42], Absorbtion Only [43], and 2D-DCGAN [44], each trained on a dataset of synthesized images in ShapeNet by the renderer of the respective model using 3D objects.

FID Score Li *et al*. [22] evaluate the quality of generated results quantitatively using an FID score [45] with an existing voxel classification network trained on the ShapeNet dataset. They measure FID scores of MP-GAN on synthetic training data for a varying

number of projections and FID scores of VP-MP-GAN (*i.e.*, with view prediction) on the same data for a varying number of view clusters. The FID scores have an upper-bound set by an MP-GAN trained with an exact viewpoint. A lower FID score indicates a higher quality image.

Using Different Fragment Sizes Hermoza and Sipiran [28] evaluates the performance of ORGAN on different sizes of fractures of complete objects. They look at how much information the model can recover from fractures even in the case where more than half of the voxels are not present. The recovery might produce some misplaced voxels. Additional tests were run on the model when the number of missing voxels was greater than 2000. They also performed tests on real objects, which resulted in the reconstruction of some unexpected fragments. This was due to the significantly different structure of the real objects from the structure of training objects.

Comparing Performance Against Existing Models Jangid *et al*. [20] compares the performance of 3DMaterialGAN with 10 existing models of which 6 are supervised and the remaining 4 are unsupervised. For comparable evaluation of unsupervised learned features, they adopted the method by Wu *et al*. [10] and the ModelNet dataset from Chang et al's work [37]. The results show that their model outperforms the existing models on both ModelNet10 and ModelNet40 dataset even with fewer training samples than those used in the existing models. They claim that with a comparable training dataset, the performance if their model improves slightly more. They also used statistical distribution comparison of 3D moment invariants to assess the quality of generated results. Zhang *et al*. [23] takes a different approach by comparing inception scores against itself and a variety of other methods such as 3D-GAN and 3D-IWGAN, among others. Evaluation determined that Zhang's 3D-Stack-SNGAN outperformed all methods tested against.

4.4 Limitations

A common limitation and challenge found in the studied works is the lack of application to practical settings such as high resolution RGB images and arbitrary camera locations [27]. Another common issue is the inability to model concavities and other features as they often need a complete unoccluded view of the objects in images [22]. Also, as GAN systems can archive very high performance scores, sometimes research that is effective may fail to surpass other state-of-the-art systems (*e.g.*, [25]). Applications can also suffer from poor FID scores [22] as there is an apparent lack of utilization of FID scores as a metric. A potential cause of this is the limited size of datasets as FID operates off of comparison for output with ground truth training data. In Chen *et al*.'s work [25], a strong performance against a baseline was shown. However, a lack of comparable performance with the state-of-the-art was also seen.

5. ANSWERING RESEARCH QUESTIONS

In this section, we answer the research questions posed in Section 4 with respect to the findings presented in the previous section.

What commonalities can be extracted from analyzing current methods of 3D object generation using GANs? Architecturally, we found that the structure of the generator is surprisingly consistent across the studied works, although there exist some differing (*e.g.*, [25]) and novel (*e.g.*, [20]) implementations which are in the minority. We also identified the use of encoders and specific training datasets to be fairly common. 80% of the studied works used either ShapeNet or ModelNet (see Table 3). With respect to the optimization of testing, the use of discriminator networks is found to be the most common accounting for 40% of the studied works (see Section 4.2.5). There were also other proven ways of test optimizing shown by others using a 2D mask or Wasserstein loss with a gradient coefficient. Regarding evaluation methods, Jaccard Index (IoU), which accounts for 40%, is found to be the most commonly used (see Section 4.3).

With these commonalities in mind, what inferences can be made about the state of this specific application of GANs? The consistent structure of the generator used in the studied works suggests that the structure may be optimal. This would allow researchers to focus on improving other areas of the subject. The common use of encoders is an expected addition to a network due to the increases to ease of use and robustness afforded by integrating data preprocessing functionality. The frequent reuse of datasets implies the lack of build-up in quality datasets that are applicable to the problem and the immaturity of the research in GANs [2]. For data optimization, discriminator networks are used most. This may be due to the fact that GAN already has its own discriminator and there is less need for more work such as masking 2D images which needs an encoder and generator. Evaluation testing was most commonly conducted using the Jaccard Index, which is unsurprising due to its suitability as a metric for volumetric representation types, such as voxels. Its ability to provide an intersection ratio between the reconstructed object and its ground truth makes the Jaccard Index an excellent choice for quantifiably evaluating reconstruction accuracy with minimal additional information.

What limitations can be identified on a general scale? It was identified that the resolution of generated 3D objects was heavily constrained (*e.g.*, being infeasible for practical use [21]) (see Section 4.2.1). While the generated results show promise, performance appears to be the primary limiting factor with memory requirements growing cubically relative to output resolution [19]. Segato *et al*. [21] suggest future work be put towards integrating a super-resolution network to enhance the output resolution despite the fact that the training images were significantly down-scaled during preprocessing. Additionally, the ability to have depth and breadth of training data can limit general scale as it impedes the development of improvements to correct performance limitations in the systems such as the need for unoccluded images [22].

What recommendations can be made towards future works? Based on the findings of this study, a multitude of recommendations can be made. First, future research should explore avenues of possible performance improvements. The quality of generated 3D objects at current resolutions show promise, but the increase in computational requirements due to higher resolutions prevent adoption into practical use. The integration of super-resolution networks to enhance generated outputs may be worth consideration. Han *et al*. [19] lends credence to this by discussing and classifying works employing various

techniques to reconstruct 3D objects at higher resolutions, while maintaining reasonable memory requirements. Second, future works should consider exploring possible new methods for the development of stronger training datasets. We see improvements to the training of GAN systems to be a potential catalyst for future growth. This growth could improve the quantitative performance of networks, specifically in Jaccard Index and FID score. Additionally, stronger datasets could expand on the applicability of FID scoring to more novel GAN solutions in the future. Additionally, GANs have a unique ability to generate training data for themselves in that indistinguishable generated data can be used as supplement training sets. So, continuing research in automatic generation of training data can also act as an accelerator to GAN development (*e.g.*, [46]).

6. CONCLUSION

We have presented a systematic literature review of multiple approaches using GANs for the purpose of volumetric 3D reconstruction using voxel representation in order to identify and discuss the commonalities that exist between them. Identifying these commonalities allows us to better understand the current state of the technology when applied this way as well as perhaps gives a hint towards what problems there are to solve and advancements to still make. In our review, we performed a deep dive study on 10 research papers published between 2018 and 2021. Our results highlighted the extreme effectiveness of pairing adversarial methods with the known power of neural networks [20]. Additionally, it showcased the strengths of particular datasets and the versatility in training they can provide when they present a strong breadth of training categories in their sets [23].

REFERENCES

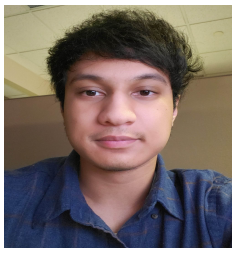
1. H. Li, Y. Zheng, X. Wu, and Q. Cai, "3D model generation and reconstruction using conditional generative adversarial network," *International Journal of Computational Intelligence Systems*, Vol. 12, 2019, pp. 697-705.
2. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2014, pp. 2672-2680.
3. J. Brownlee, "How to explore the GAN latent space when generating faces," https://machinelearningmastery.com/how-to-interpolate-and-perform-vector-arithmetic-with-faces-using-a-generative-adversarial-network/, 2019.
4. Y. Xu, X. Tong, and U. Stilla, "Voxel-based representation of 3D point clouds: Methods, applications, and its potential use in the construction industry," *Automation in Construction*, Vol. 126, 2021, No. 103675.
5. S. Lunz, Y. Li, A. Fitzgibbon, and N. Kushman, "Inverse graphics GAN: learning to generate 3D shapes from unstructured 2D data," https://arxiv.org/abs/2002.12674, 2020.
6. C. Kitchenham, "Guidelines for performing systematic literature reviews in software engineering," Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 2007.

7. A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, Vol. 35, 2018, pp. 53-65.
8. J. Zhu, P. Krähenbühl, E. Shechtman, and A. Efros, "Generative visual manipulation on the natural image manifold," in *Proceedings of European Conference on Computer Vision*, 2016, pp. 597-613.
9. J. Liu, F. Yu, and T. Funkhouser, "Interactive 3D modeling with a generative adversarial network," in *Proceedings of International Conference on 3D Vision*, 2017, pp. 126-134.
10. J. Wu, C. Zhang, T. Xue, W. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," in *Proceedings of the 30th Conference on Neural Information Processing Systems*, 2016, pp. 82-90.
11. A. Aggarwal, M. Mittal, and G. Battineni, "Generative adversarial network: An overview of theory and applications," *International Journal of Information Management Data Insights*, Vol. 1, 2021, No. 100004.
12. Y. Yu, Z. Huang, F. Li, H. Zhang, and X. Le, "Point encoder GAN: A deep learning model for 3D point cloud inpainting," *Neurocomputing*, Vol. 384, 2020, pp. 192-199.
13. Y. Chen, F. Shi, A. Christodoulou, Y. Xie, Z. Zhou, and D. Li, "Efficient and accurate MRI super-resolution using a generative adversarial network and 3D multi-level densely connected network," in *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018, pp. 91-99.
14. G. Ye, Z. Zhang, L. Ding, Y. Li, and Y. Zhu, "GAN-based focusing-enhancement method for monochromatic synthetic aperture imaging," *IEEE Sensors Journal*, Vol. 20, 2020, pp. 11 484-11 489.
15. Q. Ma, J. Yang, A. Ranjan, S. Pujades, G. Pons-Moll, S. Tang, and M. Black, "Learning to dress 3D people in generative clothing," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6469-6478.
16. Y. Jin, J. Zhang, M. Li, Y. Tian, and H. Zhu, "Towards the high-quality anime characters generation with generative adversarial networks," in *Proceedings of the Machine Learning for Creativity and Design Workshop at NIPS*, 2017.
17. P. Shamsolmoali, M. Zareapoor, E. Granger, H. Zhou, R. Wang, M. Celebi, and J. Yang, "Image synthesis with adversarial networks: A comprehensive survey and case studies," *Information Fusion*, 2021, pp. 126-146.
18. E. Smith and D. Meger, "Improved adversarial systems for 3D object generation and reconstruction," in *Proceedings of Conference on Robot Learning*, 2017, pp. 87-96.
19. X. Han, H. Laga, and M. Bennamoun, "Image-based 3D object reconstruction: state-of-the-art and trends in the deep learning era," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 43, 2021, pp. 1578-1604.
20. D. Jangid, N. Brodnik, A. Khan, M. Echlin, T. Pollock, S. Daly, and B. Manjunath, "3DMaterialGAN: Learning 3D shape representation from latent space for materials science applications," https://arxiv.org/abs/2007.13887, 2020.
21. A. Segato, V. Corbetta, M. D. Marzo, L. Pozzi, and E. D. Momi, "Data augmentation of 3D brain environment using deep convolutional refined auto-encoding alpha GAN," *IEEE Transactions on Medical Robotics and Bionics*, Vol. 3, 2020, pp. 269-272.

22. X. Li, Y. Dong, P. Peers, and X. Tong, "Synthesizing 3D shapes from silhouette image collections using multi-projection generative adversarial networks," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5535-5544.
23. H. Zhang, C. Qiu, C. Wang, B. Wei, Z. Yu, H. Zheng, and J. Li, "Learning spectral normalized adversarial systems with stacked structure for high-quality 3D object generation," *Concurrency and Computation: Practice and Experience*, Vol. 33, 2021.
24. Q. Wan, Y. Li, H. Cui, and Z. Feng, "3D-Mask-GAN: Unsupervised single-view 3D object reconstruction," in *Proceedings of the 6th International Conference on Behavioral, Economic and Socio-Cultural Computing*, 2019, pp. 1-6.
25. Y. Chen, M. Garbade, and J. Gall, "3D semantic scene completion from a single depth image using adversarial training," in *Proceedings of IEEE International Conference on Image Processing*, 2019, pp. 1835-1839.
26. Y. Chen, D. Tan, W. Cheng, and K. Hua, "3D object completion via class-conditional generative adversarial network," in *Proceedings of International Conference on Multimedia Modeling*, 2019, pp. 54-66.
27. G. Yang, Y. Cui, S. Belongie, and B. Hariharan, "Learning single-view 3D reconstruction with limited pose supervision," in *Proceedings of European Conference on Computer Vision*, 2018, pp. 86-101.
28. R. Hermoza and I. Sipiran, "3D reconstruction of incomplete archaeological objects using a generative adversarial network," in *Proceedings of Computer Graphics International*, 2018, pp. 5-11.
29. S. Song, F. Yu, A. Zeng, A. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1746-1754.
30. T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401-4410.
31. I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," https://arxiv.org/abs/1704.00028, 2017.
32. M. Mirza and S. Osindero, "Conditional generative adversarial nets," https://arxiv.org/abs/1411.1784, 2014.
33. M. Gadelha, S. Maji, and R. Wang, "3D shape induction from 2D views of multiple objects," in *International Conference on 3D Vision*, 2017, pp. 402-411.
34. N. Kodali, J. Hays, J. D. Abernethy, and Z. Kira, "On convergence and stability of GANs," https://arxiv.org/abs/1705.07215, 2018.
35. A. Larsen, S. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," https://arxiv.org/abs/1512.09300, 2015.
36. J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "CVAE-GAN: Fine-grained image generation through asymmetric training," https://arxiv.org/abs/1703.10155, 2017.
37. A. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3D model repository," https://arxiv.org/abs/1512.03012, 2015.
38. Z. Wu, S. Song, A. Khosla, X. Tang, and J. Xiao, "3D shapeNets for 2.5D object recognition and next-best-view prediction," https://arxiv.org/abs/1406.5670v2, 2014.

39. S. Liu, Y. Hu, Y. Zeng, Q. Tang, B. Jin, Y. Han, and X. Li, "See and think: Disentangling semantic scene completion," in *Processing of the 32nd Conference on Neural Information Processing Systems*, 2018.
40. T. Shubham, A. Alexei, and M. Jitendra, "Multi-view consistency as supervisory signal for learning shape and pose prediction," https://arxiv.org/abs/1801.03910, 2018.
41. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computervision and Pattern Recognition*, 2016, pp. 2818-2826.
42. M. Gadelha, S. Maji, and R. Wang, "3D shape induction from 2D views of multiple objects," https://arxiv.org/abs/1612.05872, 2016.
43. P. Henzler, N. Mitra, and T. Ritschel, "Escaping Plato's cave using adversarial training: 3D shape from unstructured 2D image collections," https://arxiv.org/abs/1811.11606, 2018.
44. A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," https://arxiv.org/abs/1511.06434, 2016.
45. M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale updaterule converge to a local nash equilibrium," in *Proceedings of the 13th International Conference on Neural Information Processing Systems*, 2017, p. 6626-6637.
46. B. Cao, H. Zhang, N. Wang, X. Gao, and D. Shen, "Auto-GAN: Self-supervised collaborative learning for medical image synthesis," in *Proceedings of AAAI Conference on Artificial Intelligence*, Vol. 34, 2020, pp. 10 486-10 493.

Riley Byrd completed his MS in Computer Science at Oakland University's School of Engineering and Computer Science. He received a BS in Computer Science from Michigan State University's College of Engineering in 2020. He worked as both a Student Information Technician and a Student IT Supervisor the years he attended MSU. His research interests include applications of artificial intelligence, robotics and computer graphics.



Kulin Damani is a Controls Engineer for Link Engineering. He is working on obtaining his Masters of Science in Software Engineering and IT at Oakland University. He received a Bachelors of Science in Computer Engineering as well as Electrical Engineering from Oakland University in 2019. His interests are data science, data analytics, and AI.

Hongjia Che is a graduate student in the Department of Computer Science and Engineering at Oakland University. He received a bachelor degree in Computer Science at the Oakland University in 2020. He is currently working as a Teaching Assistant for the Computer Science Department. His research interests include android development and artificial intelligence.

Anthony Calandra completed his MS in Computer Science at Oakland University. He also works as a Software Engineer at Honeywell Aerospace. His research interests include mobile robotic localization, medical imaging, and data science.

Dae-Kyoo Kim is a Professor in the Department of Computer Science and Engineering at Oakland University. He received a Ph.D. in Computer Science from Colorado State University in 2004. He also worked as a Technical Specialist at the NASA Ames Research Center in 2002. His research interests include software engineering, software security, and data modeling in IoT and smart grids.