Learning Universal Vector Representation for Objects of Different 3D Euclidean formats

Chengzhi Wu

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
chengzhi.wu@kit.edu

Abstract

We present a method for learning universal vector representations out of 3D objects represented in different data formats. A newly proposed switching mechanism is used in the design of neural network architecture. During the learning process, the encoder for one specific format also learns to perceive the object from the perspective of other formats, hence the learned universal representation contains richer information. With the learned universal representation, it would be possible to "translate" between different 3D shape formats of the input object since they share similar embedding of 3D information. Higher performance can also be achieved for the 3D data synthetic tasks with this method.

1 Introduction

1.1 Latent representation of 3D data

Depending on the measuring method and the processing and storing rules of information, 3D objects may have various representing formats in the real-world. On the Euclidean side, they may be represented as RGB-D images, multi-view images or volumetric data. On the Non-Euclidean side, they may be represented as point clouds or meshes. However, no matter in which format store the 3D information of the object, when it comes to the computer vision tasks, e.g. detection, segmentation, or even other generative tasks, the target 3D object will usually need to be converted into a latent representation first for further computation.

Before the surge of deep learning, it was common to use classical mathematical algorithms to get those 3D shape latent representations (or, 3D shape descriptors). This computation process usually involves strict mathematical formulas and deductions to get rule-based representations, e.g. Laplacian spectral eigenvectors [15], or heat kernel signature [17]. Thanks to the development of deep learning algorithms, the performance of some computer vision tasks, especially in the detection and segmentation domain [7], have been boosted. During the training of those neural networks, latent representations of input have already been generated implicitly. Although this learning process has been regarded as a black box at earlier years, researches in the visualization of learned latent representations have been conducted [24]. Throughout the computer vision learning history, a better method for learning the latent representations leads to better performance on those tasks.

1.2 Universal vector representation

Learning an universal vector representation touches on two long-standing and important questions in computer vision: how do we represent 3D objects in a vector space and how do we recognize this representation from images. [6] believed that a good vector representation for objects must satisfy two criteria: it must be (1) generative in 3D; (2) predictable from 2D. In this report, we learn universal representations for 3D objects of different formats by leveraging the advantages of deep learning algorithms. For simplicity, we are only investigating Euclidean data in this report.

On the one hand, the vector representation can be learned from different data formats such as multi-view images and volumetric data; On the other hand, it can be inferred during the training process of different neural networks designed for different machine vision tasks. For analytical tasks, especially for the classic classification tasks, vector representations will usually be learned before the last several fully connected layers. These vector representations are sometimes referred as bottleneck features. Those bottleneck features can be further adapted for other tasks, as it is done in transfer learning. For synthetic tasks, typical generative models are AE/VAE [11] and GAN [8]. They learn the mappings between the latent space and the real-world data space, thus reconstructions from latent representations are possible. Theoretically, if we can learn universal representations that contain both view information and geometry information, better synthetic results may be achieved. Hence those generative models may be modified for learning universal representations in our case and may also be used as a verifier to indicate the performance.

From another perspective, the process may also be regarded as data compression and the richness of its implicitly stored feature information is of pivotal importance. Since the learned universal vector representations can be used not only for synthetic tasks, but also for analytical tasks, we are also expecting higher performance in regular machine vision tasks like classification with them. Since we want to merge the information from different data formats, the resolution of data should also be considered. It would be apparently inappropriate to have a fixed-size latent vector to represent objects of different resolutions, even under the same format. Hence, the main idea of this report is to learn a fixed length of vector representation for an object of a specific category, under certain resolution limitations of different data formats.

2 Related Work

2.1 Learning representations (encoders)

Although latent representations are also learned in analytical tasks, they have been seldom specifically explored. There are numerous papers using various network architectures for 3D machine vision tasks. A typical one is VoxNet[13], which was the first to use 3D convolution operations to learn features from volumetric data. Its subsequent work of multi-level 3D CNN [5] learns multi-scale spatial features by considering multiple resolutions of the voxel input. Regarding the multi-view images format, a typical method is MVCNN [16]. It uses a parameter sharing network to encode images of one object from different views, followed by a view pooling layer before the last several fully connected layers. A subsequent work of GVCNN [4] groups all the views before encoding. Each group uses one separate parameter sharing network to encode this group of images, then a group fusion operation is defined in the latter step.

Methods combing the information from both multi-view images and volumetric data have also been proposed. For example, Qi et al. [14] proposed to use multi-resolution filtering in 3D for multi-view CNNs, as well as using subvolume supervision for auxiliary training. Another example is FusionNet [10], which is a fusion of three different networks: two VoxNets and one MVCNN. The three networks fuse at the score layers where a linear combination of scores is taken before finding the class prediction. Voxelized CAD models are used for the first two networks and 2D projections are used for latter network.

2.2 Generating from representations (decoders)

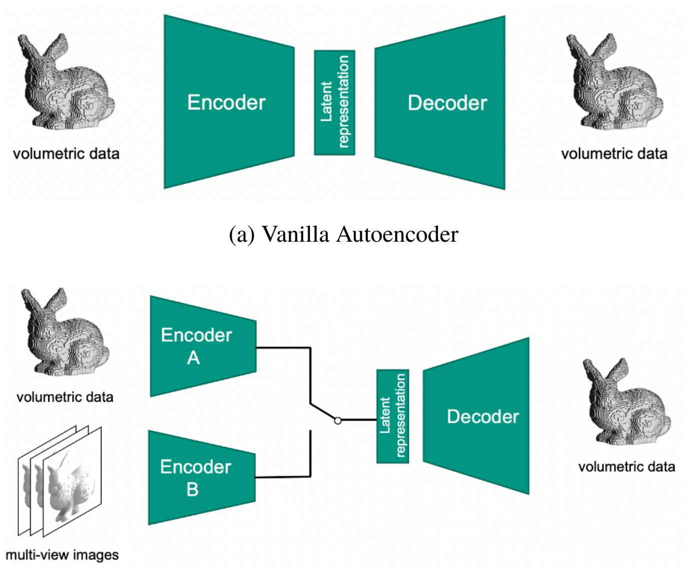
Unlike analytical tasks, latent representation matters a lot to synthetic tasks. There are mainly two types of deep generative models nowadays: AE/VAE [11] and GAN [8]. Based on those two frameworks, various methods have been proposed to learn latent representations from 3D data and to reconstruct back to them. ShapeNet [23] used a reverse VoxNet, i.e. a decoder, to reconstruct 3D shapes from a latent representation which was learned from depth maps. The dataset they created is also being widely used for 3D machine vision tasks nowadays. Girdhar et al. [6] used AE directly to encode and decode 3D shapes. With the learned latent representation from volumetric data, they proposed a TL-embedding network which forces another encoder to learn a exactly the same latent representation from corresponding images. This makes it possible to generate 3D shapes from images. VAE has also been used in a similar way for the 3D shape learning in other paper [2].

View information from images has also been widely investigated for 3D shape reconstruction. Choy et al. [3] proposed a framework named 3D-R2N2 to reconstruct 3D shapes from single- or multi-view images. By leveraging the power of Long Short-Term Memory(LSTM), they discovered that the reconstruction is incrementally refined as the network sees more views of the object. Some other papers also have used view information as auxiliary constraints for the training of their 3D AEs. Tulsiani et al. [18] trained an additional pose CNN to add an additional consistency loss between the inferred depth image from a perspective and its ground truth. This inferred ray-trace pooling view has also been used in the adversarial part of [9] for weakly supervision.

Methods used GAN for 3D shape generation have also been proposed. 3D-GAN [21] makes it possible to generate novel and relatively highly realistic 3D model in the unsupervised way. It introduced three loss functions regarding image encoder, generator and discriminator respectively. The update of all components in the framework is also possible. Besides, visualizing the representation vector, interpolation, arithmetic have been conducted to analyse the vector representations. Liu et al. [12] adapted this idea and proposed an interactive modeling framework that can generate realistic volumetric data with edit and especially defined snap operations. Semantic information has been used in Global-to-Local GAN (G2LGAN) [19] and SAGNet [22] to improve the synthesis quality. G2LGAN also proposed a part refiner to refine the individual semantic part output from local GANs. While showing that segmented information from 3D data can be embedded into the latent space, their work does not include too much discussion of the latent space and its connection with other data formats like image or common voxel.

3 Methodology

Although both AE and GAN were developed for data synthesis tasks by using neural networks, they are quite different in their kernel ideas. AEs use real-world data as input. An encoder-decoder structure network is used to encode the input into latent representation, and subsequently reconstruct it back from the latent representation. The most important loss here is the reconstruction loss. With a well-trained decoder, it is possible to reconstruct the object with a well-learned latent representation. An unsolved question here is how can we force the latent representation to be meaningful. GANs are totally different from AEs since they do not use real-world data as input directly. Instead, they train a generator,



(a) Vanilla Autoencoder

(b) Switch-Autoencoder

Figure 2.1: An illustration of the (a) vanilla Autoencoder (AE) and the proposed (b) Switch-Autoencoder (SAE). AE only takes volumetric data as input, while SAE takes input from both image data and volumetric data, using a switch to randomly choose the learning source. The feature maps/vectors learned inside the network may be regarded as latent representations.

which is similar to the decoder in AEs, on the latent space directly. Generated data will be processed into a discriminator to classify it is generated or from the real world. The whole training process is essentially the competition between the generator and the discriminator. For GANs, the distribution of the latent representation is usually pre-defined as a Gaussian, but how to disentangle the features in the latent space is still a tough question.

In our case, since we are interested in learning a universal representation from 3D data of multi-formats, the original 3D information should be fully utilized. Hence here we adopt the AE architecture to learn latent representations. GAN

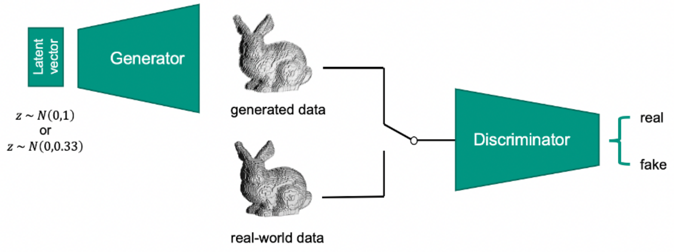


Figure 3.1: The basic structure of a generative adversarial network (GAN). It trains a generator to decode the latent vector representation to a 3D shape, by using a discriminator to force the generator to generate shapes as real as possible.

is great for generating the 3D data, it may be combined with the AE part for better reconstruction in the future step.

3.1 Vanilla Autoencoder

Firstly, we started testing our idea with a simple Autoencoder. As shown in Figure 2.1(a), it is just a normal AE but with 3D convolutions. The loss of this network is the reconstruction loss. There are different ways to compute the reconstruction loss including MSE, Cross Entropy, and IoU. IoU is more like an indicator and does not provide smooth gradient. MSE is mainly used for preliminary tests. In our case, we use the cross entropy as loss function. The output before the last layer has been rectified to a range from 0 to 1. The dataset we are using here is the ShapeNet [23]. It provides a wide variety of real 3D objects, which makes the data-driven learning and analysing of the latent representation possible and promising. The synthesis result from this architecture may be regarded as the baseline of performance.

3.2 The Switch-Autoencoder

The multi-view images data is added to the input side in this setting of experiment. Here, we propose a Switch-Autoencoder (SAE) for universal latent representation learning. We train two encoders separately for the voxel input and the image sequence input. A switch is attached before the decoder. During the training, the network randomly selects the encoded output from one encoder as the latent representation, then inputs it to the decoder. This operation of switching between encoders continues during the whole end-to-end training. In the TL-embedding network proposed in [6], the image encoder is forced to learn the same embedding from that of the voxel encoder, hence the image encoder does not contribute to the improvement of the generator. Unlike TL-embedding network, in our case, both encoders learn to perceive the object from the perspective of the other format, hence both encoders contribute to the improvement of the generator.

The structure of proposed SAE is shown in Figure 2.1(b). For the image encoder (encoder B), we use an architecture that is similar to multi-view CNN [16]. Each view is encoded with a parameter-sharing network, followed by a view pooling layer. Then it will be passed through several additional fully connected layers to get the final latent vector representation. Here, we also use a network with residual blocks in the image view encoder.

3.3 GAN

In order to improve the synthesis quality, the framework of GAN may be integrated here. In this report, we are focusing on examining the quality of generated shapes from GAN with normal Gaussian vector input. Its basic structure is shown in Figure 3.1. The discussion and experiments of combining AE/VAE and GAN is in the scope of our next step.

There are several non-negligible problems in the vanilla GANs. When training the standard GAN, the loss of the discriminator and generator can oscillate gradually, which make the training process unstable. Besides, vanilla GANs also have the problem of mode collapse, which produces limited varieties of samples. Here, we use the Wasserstein GAN(WGAN) [1] with gradient penality [20], which has two main benefits: (1) improved stability of training process; (2) a meaningful loss metric that correlates with generators convergence and sample equality. We believe that merging the WGAN in the framework will improve the quality of object generation.

4 Experimental Results

4.1 Vanilla Autoencoder

Figure 4.1 shows some results from the vanilla Autoencoder. From it we can tell that the network is able to reconstruct the input 3D shapes from learned latent representations. Besides, the features of different types of chairs have also been well captured.

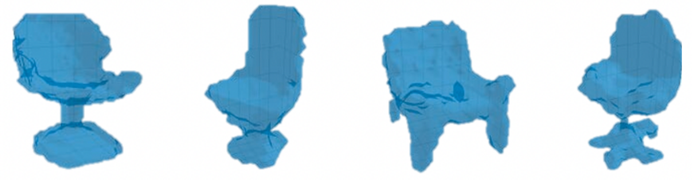
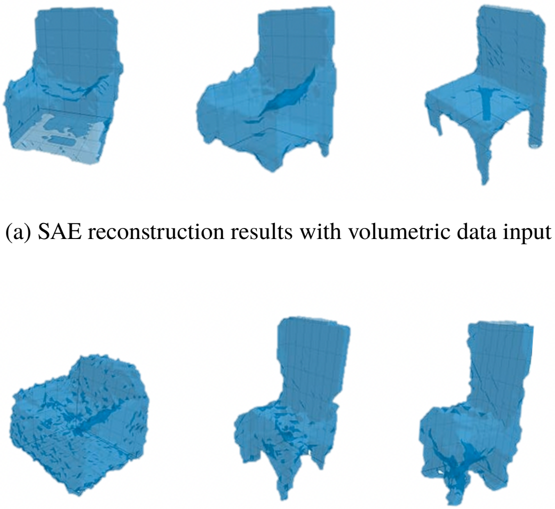


Figure 4.1: Reconstruction results of chair objects from the vanilla Autoencoder.

4.2 The Switch-Autoencoder

Some reconstruction results from SAE is given in Figure 4.2. The up row shows the results with volumetric data as the test input. The bottom row shows the results with multi-view image sequence as the test input. From it we can see that the voxel-encoder still preserves a relatively high quality, while the image-encoder also captures decent 3D information. The sharp areas are difficult for the image-encoder, as can be observed from the generated chair legs.

Overall, comparing with the results from only one format source, objects generated from universal representations with both format sources look better. The generated chairs are usually less rough.



(a) SAE reconstruction results with volumetric data input

(b) SAE reconstruction results with image data input

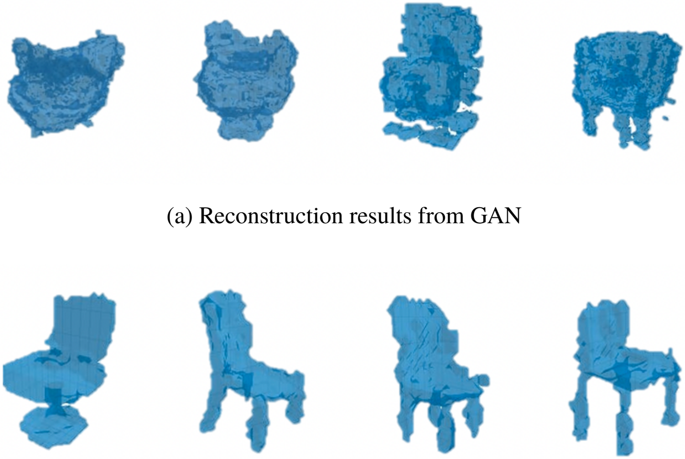
Figure 4.2: Reconstruction results of chair objects from the proposed Switch-Autoencoder, using (a) volumetric data or (b) image data as test input, respectively.

4.3 GAN

GANs are unsupervised learning algorithms that use a supervised loss as part of the training. So their results are expected to be as good as the ones from an AE. Figure 4.3(a) gives some results from a vanilla GAN. We can observe that the vanilla GAN only captures very basic bulky features of chairs but fails on the details, even using a relatively higher resolution. Another disadvantage of the vanilla GAN is that, it can fall into mode collapse easily. In this case, the discriminator is trained too well to classify generated models too easily, thus the generator does not learn anything at all.

As discussed in Section 3, we are adopting the WGAN method with gradient penalty to overcome the aforementioned problems. Since we are using WGAN, the last sigmoid layer of the discriminator has been removed. No log operations were used for both losses. Besides gradient penalty, noise was added by doing interpolation between generated and real data before feeding into the discriminator. Figure 4.3(b) gives some optimal results obtained in our experiments. The latent representations we used were sampled from a distribution of $N(0, 0.33)$. As can be observed from the figures, although the generated objects are not extreme smooth, they are already in decent chair-like shapes.

Experiments with other settings have also been carried out. For example, Figure 4.4(a) gives the results of using original sigmoid layer instead of tanh for the last layer of generator. The model may generate lots of floating artifacts shortly after the training begins. In order to reduce the memory consumption for future architecture update, we tried to half the number of feature maps we used between the layers. From Figure 4.4(b) we can observe that obviously the results are

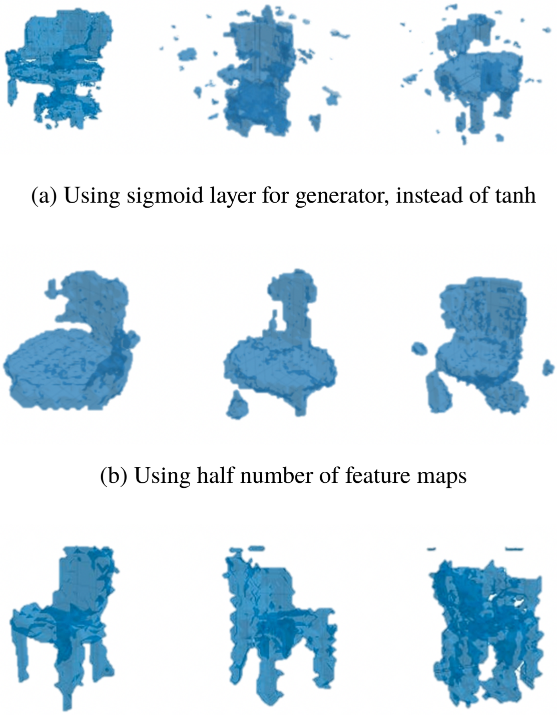


(a) Reconstruction results from GAN

(b) Reconstruction results from WGAN-GP

Figure 4.3: Reconstruction results of chair objects from (a) vanilla GAN and (b) WGAN-GP. The vanilla GAN only captures very basic bulky features of chairs but fails on the details, even using a higher resolution. WGAN-GP can already generate decent chair-like shapes.

not promising anymore. Regarding the initial parameter distribution of the latent representations, experiments have been done with the more often used distribution of $N(0, 1)$, results are shown in Figure 4.4(c). Apparently the generated objects are more noisy.



(a) Using sigmoid layer for generator, instead of tanh

(b) Using half number of feature maps

(c) Using a latent vector distribution of $N(0, 1)$, instead of $N(0, 0.33)$

Figure 4.4: Reconstruction results of chair objects from WGAN-GP with other different settings. (a) For the last layer of the generator, using sigmoid instead of tanh. (b) Using half number of feature maps in the network. (c) The latent representations are sampled from a distribution of $N(0, 1)$, instead of $N(0, 0.33)$.

4.4 AE/VAE-GAN and more

In previous subsections, we have proven that our AE model and GAN model are working. To learn a better universal representation and achieve better synthesis performance, we may combine those two models since the decoder part in AE is exactly the generator part in GAN. However, during the actual testing, this idea never worked if they are straightforwardly combined. The main problem of this idea is that the learned universal representation with AE does not naturally follow a Gaussian distribution, while it is a mandatory requirement for GAN as the input. Hence, Variational Autoencoder (VAE) should be used here for the integration. For the encoder of an Autoencoder, each input is mapped directly to one point in the latent space, which leads to the discontinuous latent space and huge gaps between groups of similar points from the input space. In a variational autoencoder, each input is instead mapped to a multivariate normal distribution around a point in the latent space, which makes a continuous latent space. Continuous latent space also makes the generation of new 3D object possible and the analysis of latent space easier.

On the other hand, this report is mainly about the learning process of universal representations. Reconstructed objects are used to validate the effectiveness of proposed method. In order to make it more illustrative, experiments regarding the investigations in the latent space should be carried out in future. For example, not only for the synthesis tasks, but also for the analytical tasks including object classification.

5 Conclusion and Outlook

In this report, we proposed a switch autoencoder method to learn universal latent representations for 3D object with multiple-formats input. Synthesis experiments have been carried out to validate the effectiveness of the proposed method. With the learned universal representation, decoders can generate 3D objects of better quality. The next step of our future experiments is to make VAE and VAE-GAN work, with which the interpretability of the learned latent representation may be explored. As discussed in Section 4.4, more experiments will be done regarding the latent space, e.g. similarity search or shape interpolation. In the future, other 3D formats like point cloud may be included. Semantic information may also be used here for better interpretable latent representation learning.

References

[1] Martín Arjovsky, Soumith Chintala, and L. Bottou. “Wasserstein GAN”. In: *ArXiv* abs/1701.07875 (2017).

[2] A. Brock et al. “Generative and Discriminative Voxel Modeling with Convolutional Neural Networks”. In: *ArXiv* abs/1608.04236 (2016).

[3] C. Choy et al. “3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction”. In: *European Conference on Computer Vision (ECCV)* (2016).

[4] Y. Feng et al. “GVCNN: Group-View Convolutional Neural Networks for 3D Shape Recognition”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 264–272.

[5] Sambit Ghadai et al. “Multi-Level 3D CNN for Learning Multi-Scale Spatial Features”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2019), pp. 1152–1156.

[6] Rohit Girdhar et al. “Learning a Predictable and Generative Vector Representation for Objects”. In: *European Conference on Computer Vision (ECCV)* (2016).

[7] I. Goodfellow, Yoshua Bengio, and Aaron C. Courville. “Deep Learning”. In: *Nature* 521 (2015), pp. 436–444.

[8] Ian J. Goodfellow et al. “Generative Adversarial Networks”. In: *ArXiv* abs/1406.2661 (2014).

[9] JunYoung Gwak et al. “Weakly Supervised 3D Reconstruction with Adversarial Constraint”. In: *2017 International Conference on 3D Vision (3DV)* (2017), pp. 263–272.

[10] Vishakh Hegde and R. Zadeh. “FusionNet: 3D Object Classification Using Multiple Data Representations”. In: *ArXiv* abs/1607.05695 (2016).

[11] Diederik P. Kingma and M. Welling. “An Introduction to Variational Autoencoders”. In: *Found. Trends Mach. Learn.* 12 (2019), pp. 307–392.

[12] Jerry Liu, F. Yu, and T. Funkhouser. “Interactive 3D Modeling with a Generative Adversarial Network”. In: *2017 International Conference on 3D Vision (3DV)* (2017), pp. 126–134.

[13] D. Maturana and S. Scherer. “VoxNet: A 3D Convolutional Neural Network for real-time object recognition”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2015), pp. 922–928.

[14] C. R. Qi et al. “Volumetric and Multi-view CNNs for Object Classification on 3D Data”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 5648–5656.

[15] O. Sorkine-Hornung. “Laplacian Mesh Processing”. In: *Eurographics* (2005).

[16] Hang Su et al. “Multi-view Convolutional Neural Networks for 3D Shape Recognition”. In: *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 945–953.

[17] J. Sun, M. Ovsjanikov, and L. Guibas. “A Concise and Provably Informative MultiScale Signature Based on Heat Diffusion”. In: *Computer Graphics Forum* 28 (2009).

[18] Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. “Multi-view Consistency as Supervisory Signal for Learning Shape and Pose Prediction”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 2897–2905.

[19] H. Wang et al. “Global-to-local generative model for 3D shapes”. In: *ACM Transactions on Graphics (TOG)* 37 (2018), pp. 1–10.

[20] X. Wei et al. “Improving the Improved Training of Wasserstein GANs: A Consistency Term and Its Dual Effect”. In: *ArXiv* abs/1803.01541 (2018).

[21] Jiajun Wu et al. “Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling”. In: *ArXiv* abs/1610.07584 (2016).

[22] Z. Wu et al. “SAGNet”. In: *ACM Transactions on Graphics (TOG)* 38 (2019), pp. 1–14.

[23] Zhirong Wu et al. “3D ShapeNets: A deep representation for volumetric shapes”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 1912–1920.

[24] Luisa M. Zintgraf et al. “Visualizing Deep Neural Network Decisions: Prediction Difference Analysis”. In: *ArXiv* abs/1702.04595 (2017).