Adversarial Autoencoders for Generating 3D Point Clouds

Maciej Zamorski*†[1,5], Maciej Zięba*‡[1,5], Rafał Nowak[2,5], Wojciech Stokowiec[3,5], and Tomasz Trzciński[4,5]

[1]Faculty of Computer Science and Management, Wrocław University of Science and Technology, Wrocław, Poland
[2]Institute of Computer Science, University of Wrocław, Wrocław, Poland
[3]Polish-Japanese Academy of Information Technology, Warsaw, Poland
[4]The Faculty of Electronics and Information Technology, Warsaw University of Technology, Warsaw, Poland
[5]Tooploox Ltd., Wrocław, Poland

Abstract

Deep generative architectures provide a way to model not only images, but also complex, 3-dimensional objects, such as point clouds. In this work, we present a novel method to obtain meaningful representations of 3D shapes that can be used for clustering and reconstruction. Contrary to existing methods for 3D point cloud generation that train separate decoupled models for representation learning and generation, our approach is the first end-to-end solution that allows to simultaneously learn a latent space of representation and generate 3D shape out of it. To achieve this goal, we extend a deep Adversarial Autoencoder model (AAE) to accept 3D input and create 3D output. Thanks to our end-to-end training regime, the resulting method called 3D Adversarial Autoencoder (3dAAE) obtains either binary or continuous latent space that covers much wider portion of training data distribution, hence allowing smooth interpolation between the shapes. Finally, our extensive quantitative evaluation shows that 3dAAE provides state-of-the-art results on a set of benchmark tasks.

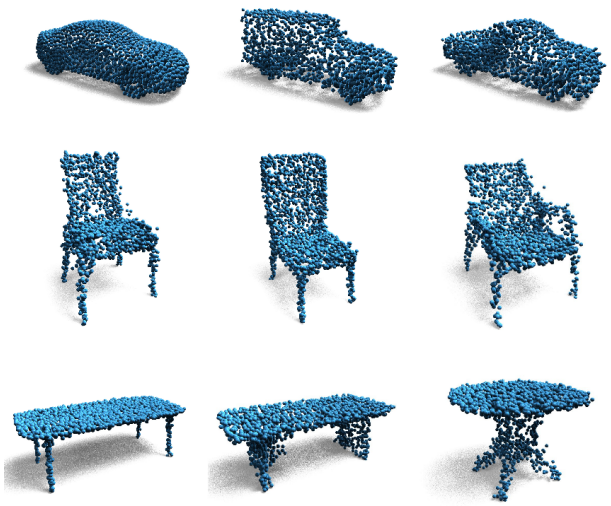

Figure 1. Synthetic point cloud samples generated by our AAE models trained with Earth Mover Distance as a reconstruction error.

1. Introduction

Creating useful and compact data representations is one of the main challenges tackled by machine learning and computer vision. Although learning various (binary and continuous) representations for 2D images has been extensively investigated [3, 9, 14, 18, 20, 21, 22], so far relatively little attention was paid to representations of three dimensional shapes and structures [1, 5, 13, 19].

As more and more sensors offer capturing depth along with other visual cues, three dimensional data points start to play a pivotal role in many real-life applications, including simultaneous localization and mapping or 3D object detection. Proliferation of devices such as RGB-D cameras and LIDARs leads to increased amount of 3D data that is being captured and analysed, e.g. by autonomous cars and robots. Since storing and processing 3D data points in their raw form quickly becomes a bottleneck of a processing system, compact and efficient representations are needed. Generative models, such as those inspired by GANs [6], can learn such representations in an unsupervised manner which reduces annotation and labeling efforts. Furthermore, those models are able at the same time and no additional cost to provide tooling for synthetic generation and augmentation of existing datasets. This, in turn, can be helpful for training

*Equal contribution
†maciej.zamorski@pwr.edu.pl
‡maciej.zieba@pwr.edu.pl

more advanced and effective machine learning algorithms, e.g. designed for 3D object detection.

Most of the contemporary works on 3D shape representations learn them using voxel-, view- and point-based projections [19, 5, 1] and we follow this approach in this paper. The main advantage of such 3D point cloud representations is their similarity to the 3D objects present in real life, however processing them can be cumbersome due to their invariance to permutation.

Due to the complex nature of 3D representations, several approaches based on deep neural networks have been proposed [1, 5, 13, 16, 19]. In [19], the authors propose an extension of a standard GAN architecture [6] that allows to generate realistic 3D shapes sampled from a manifold of latent variable space. PointNet model [13], on the other hand, provides a universal architecture to input 3D points in a permutation invariant manner and we incorporate this work in our model. [16] uses a version of Variational Autoencoder [9] to build a semantic representation of a 3D scene that is later used for relocalization in SLAM.

Perhaps the most relevant to our work is [1] where the authors propose a method for encoding and generating 3D point clouds. Their cascaded architecture consists of two separate modules: the autoencoder trained to obtain a latent representation and a corresponding generative model to transform the latent space into a 3D shape. Since those two models are trained separately, both sampling and coding representations are different and cannot be translated or interpreted within both spaces.

In this work, we address this limitation of the existing method and introduce the first end-to-end 3D point cloud generative model called 3D Adversarial Autoencoder (3dAAE) that is capable of creating smooth transitions between database objects. This functionality is available when sampling latent variable space modelled with this approach, since 3dAAE allows for much wider coverage of the training dataset manifold of 3D shapes. More precisely, thanks to introducing Earth-Mover loss into an Adversarial Autoencoder scheme connected with a PointNet module, we are able to increase the granularity of the latent space used by the generative part of the model to create point cloud. As a result, our proposed 3dAAE model offers state-of-the-art results in terms of generation and retrieval, as well as realistic generation capabilities and latent variable space arithmetics.

To summarize, in this work we present the following contributions:

- We show that variational autoencoders can be applied directly on encoding space with no significant decrease in reconstruction and generative capabilities of the model.
- We introduce adversarial autoencoders for 3D point clouds that are capable to learn any distribution on latent space.
- We show how adversarial autoencoders can solve challenging tasks, such as 3D points clustering or 3D object retrieval using binary embeddings.

Our work is organized as follows: Section 2 presents preliminary information and notation used in the paper. Section 3 outlines generative models upon which we build our proposed model. In Section 4 we introduce our approach for learning latent representations of point clouds. Section 5 contains thorough evaluation of our model with quantitive and qualitive results. The paper is summarized and concluded in Section 6.

2. Preliminaries and Notation

In this section we provide necessary definitions and present metrics used to compare similarities between two point clouds that are used for model training. We conclude this section with an explanation of the notation used in subsequent sections.

A *point cloud*, which we will denote by $\mathbf{s}$, is a $n \times 3$ matrix, where each row represent a point in a 3D euclidean space.

Earth Mover's Distance (**EMD**): introduced in [15] is a metric between two distributions based on the minimal cost that must be paid to transform one distribution into the other. For two equally sized subsets $\mathbf{s_1} \subseteq R^3$, $\mathbf{s_2} \subseteq R^3$, their EMD is defined as:

$$EMD(\mathbf{s_1}, \mathbf{s_2}) = \min_{\phi:\mathbf{s_1}\to\mathbf{s_2}} \sum_{x\in\mathbf{s_1}} \|x - \phi(x)\|_2 , \qquad (1)$$

where ϕ is a bijection.

Chamfer pseudo-distance (**CD**): measures the squared distance between each point in one set to its nearest neighbor in the other set:

$$CD(\mathbf{s_1}, \mathbf{s_2}) = \sum_{x\in\mathbf{s_1}} \min_{y\in\mathbf{s_2}} \|x - y\|_2^2 + \sum_{y\in\mathbf{s_2}} \min_{x\in\mathbf{s_1}} \|x - y\|_2^2 . \qquad (2)$$

In contrast to EMD which is only differentiable almost everywhere, CD is fully differentiable. Additionally, CD is computationally less requiring.

In the rest of our work, we use bold lower-case letters, such as $\mathbf{s}$ to represent point clouds and lower-case letter, such as x, to represent vectors. Finally, upper-case italic letters such as E, G or D denote model components.

3. Methods

3.1. Variational Autoencoders

Variational Autoencoders (**VAE**) are the generative models that are capable of learning approximated data distribution by applying variational inference [9].

We consider the latent stochastic space z and optimize the upper-bound on the negative log-likelihood of x:

$$\begin{aligned}\mathbb{E}_{x\sim p_d(x)}[-\log p(x)] <& \mathbb{E}_x[\mathbb{E}_{z\sim q(z|x)}[-\log p(x|z)]] \\ &+ \mathbb{E}_x\left[KL\left(q(z|x)\|p(z)\right)\right] \\ =& \text{ reconstruction } + \text{ regularization}, \end{aligned} \tag{3}$$

where KL is the Kullback–Leibler divergence, $p_d(x)$ is the empirical distribution, $q(z|x)$ is the variational posterior (the encoder E), $p(x|z)$ is the generative model (the generator G) and $p(z)$ is the prior. In practical applications $p(x|z)$ and $q(z|x)$ are parametrized with neural networks and sampling from $q(z|x)$ is performed by so called reparametrization trick. The total loss used to train VAE can be represented by two factors: reconstruction term, that is ℓ_2 norm taken from the difference between sampled and reconstructed object if $p(x|z)$ is assumed to be normal distribution and regularization term that forces z generated from $q(z|x)$ network to be from a prior distribution $p(z)$.

3.2. Adversarial Autoencoders

The main limitation of VAE models is that regularization term requires particular prior distribution to make $KL(\cdot\|\cdot)$ divergence tractable. In order to deal with that limitation authors of [12] introduced Adversarial Autoencoders (**AAE**) that utilize adversarial training to force a particular distribution on z space. The model assumes that an additional neural network - discriminator D, which is responsible for distinguishing between fake and true samples, where the true samples are sampled from assumed prior distribution $p(z)$ and fake samples are generated via encoding network $q(z|x)$. The adversarial part of training can be expressed in the following way:

$$\begin{aligned}\min_E \max_D V(E,D) = & \mathbb{E}_{z\sim p(z)}[\log D(z)] \\ &+ \mathbb{E}_{x\sim p_d(x)}[\log\left(1 - D(E(x))\right)]. \end{aligned} \tag{4}$$

The training procedure is characteristic for GAN models and is performed by alternating updates of parameters of encoder E and discriminator D. The parameters of discriminator D are updated by minimizing the $L_D = -V(E,D)$ and the parameters of encoder E and generator G are optimized by minimizing the reconstruction error together with $V(E,D)$: $L_{EG} = reconstruction + V(E,D)$. In practical applications, the stated criterion can be substituted with so called Wasserstein criterion [7].

Learning prior on z latent space using adversarial training has couple of advantages over standard VAE approaches [12]. First of all, the data examples coded with the encoder exhibits sharp transitions indicating that the coding space is filled which is beneficial in terms of interpolating on the latent space. Secondly, there are no limitation for the distribution that is adjusted to z space.

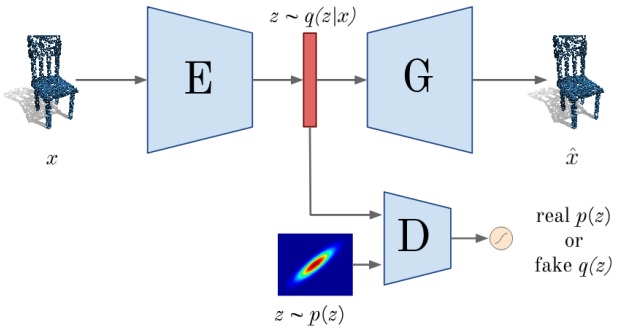

Figure 2. 3dAAE model architecture that extends AE with an additional decoder D. The role of the decoder is to distinguish between true samples generated from $p(z)$ and fakes delivered by the encoder E. Encoder is trying to generate artificial samples, that are difficult to be distinguished by the discriminator.

4. Adversarial Autoencoders for 3D point clouds

The superior generative power of GAN models [11] over classical autoencoders in terms of generating artificial images is mainly caused by the difficulties in defining good distance measure in data (pixel) space that is essential for the encoders. Thanks to the distance measures, like Chamfer of Earth-Mover, this limitation is no longer observed for 3D point clouds. Therefore, we introduce VAE and AAE models for 3D point clouds and call them **3dVAE** and **3dAAE**, respectively.

VAE for 3D point clouds (**3dVAE**) assumes an additional $KL(\cdot\|\cdot)$ regularization term in training the AE model that enforces latent space **z** to be from the assumed prior. The training criterion for the model is composed of reconstruction error defined with Earth-Mover distance between original and reconstructed samples, and $KL(\cdot\|\cdot)$ distance between samples generated by encoder and the samples generated from prior distribution. Samples from the encoder are obtained by application of reparametrization trick on the last layer. To balance the gap caused by the orders of magnitude between the components of the loss we scale the Earth-Mover component by multiplying it by the scaling parameter λ.

Due to the limitations of VAE listed in the previous section, namely narrow spectrum of possible priors and worse distribution adjustment, we propose the adversarial approach adjusted to be applied to 3D point clouds (**3dAAE**). The scheme of adversarial autoencoder for 3D point clouds is presented in Figure 2. The model is composed of an encoder E that is represented by PointNet [13]

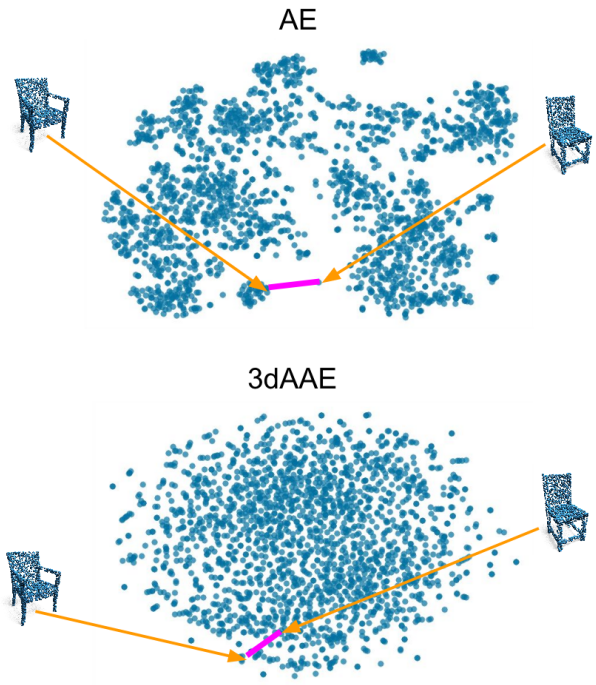

Figure 3. t-SNE plot for the latent space obtained from AE and 3dAAE models (chair category). One can notice the interpolation gap between two chairs for AE. For encodings obtained from 3dAAE model this phenomenon is not observed and the latent variable space is much more dense which allows for smooth transition within the space.

architecture and transforms 3D points into latent space z. The latent coding z is further used by generator G to reconstruct or generate 3D point clouds. To train the assumed prior distribution $p(z)$ we utilize a discriminator D that is involved in the process of distinguishing between true samples generated from prior distribution $p(z)$ and fake samples obtained from encoder E. If z is either normal distribution or Mixture of Gaussians we apply reparametrization trick to the encoder E to obtain z samples.

The model is trained in the adversarial training scheme describe in details in Section 3.2. As a reconstruction loss we take Earth-Mover loss described in Section 2. For the GAN part of training we utilize Wasserstein criterion.

In Figure 3 we present the 2D visualization of the coding space for AE and 3dAAE methods. For 3dAAE model we can observe, that the encoding are clustered consistently to the prior distribution. For the AE model we can find the gaps in some spaces that may lead in poor interpolation results.

Contrary to the stacked models presented in [1] our model is trained in end-to-end framework. The latent coding space that is used for both representation and sampling purposes. Thanks to the application of adversarial training we obtain data codes that are consistent with the assumed prior distribution $p(z)$.

5. Evaluation

In this section we describe experimental results of the proposed generative models in various tasks including 3D points reconstruction, generation, binary representation and clustering.

5.1. Metrics

Following the methodology for evaluating generative fidelity and samples diversification provided in [1] we utilize the following criteria for evaluation: Jensen-Shannon Divergence, Coverage and Minimum Matching Distance.

Jensen-Shannon Divergence (**JSD**): measure of distance between two empirical distributions P and Q, defined as:

$$JSD(P||Q) = \frac{KL(P||M) + KL(Q||M)}{2}, \qquad (5)$$

where $M = \frac{P+Q}{2}$ and $KL(\cdot||\cdot)$ – Kullback-Leibler Divergence[10].

Coverage (**COV**): measure of generative capabilities in terms of richness of generated samples from the model. For point cloud sets **A** and **B** coverage is defined as a fraction of point clouds in **B** that are in the given metric the nearest neighbor to some point cloud in **A**.

Minimum Matching Distance (**MMD**): Since COV only takes the closest point clouds into account and does not depend on distance between the matchings additional metric was introduced. For point cloud sets **A** and **B** MMD is a measure of similarity between point clouds in **A** to those in **B**.

Both COV and MMD can be calculated using Chamfer (**COV-CD**, **MMD-CD**) and Earth-Mover (**COV-EMD**, **MMD-EMD**) distances, respectively. For completeness we report all possible combinations.

5.2. Network architecture

In all of our experiments we use the following network architectures:

- Encoder (E) is a PointNet network composed of five *conv1d* layers, one fully-connected layer and two separate fully-connected layers for reparametrization trick. ReLU activations are used for all except the last layer used for reparametrization.
- Generator (G) is fully-connected network with 5 layers and ReLU activations except the last layer.
- Discriminator (D) is fully-connected network with 5 layers and ReLU activations except the last layer.

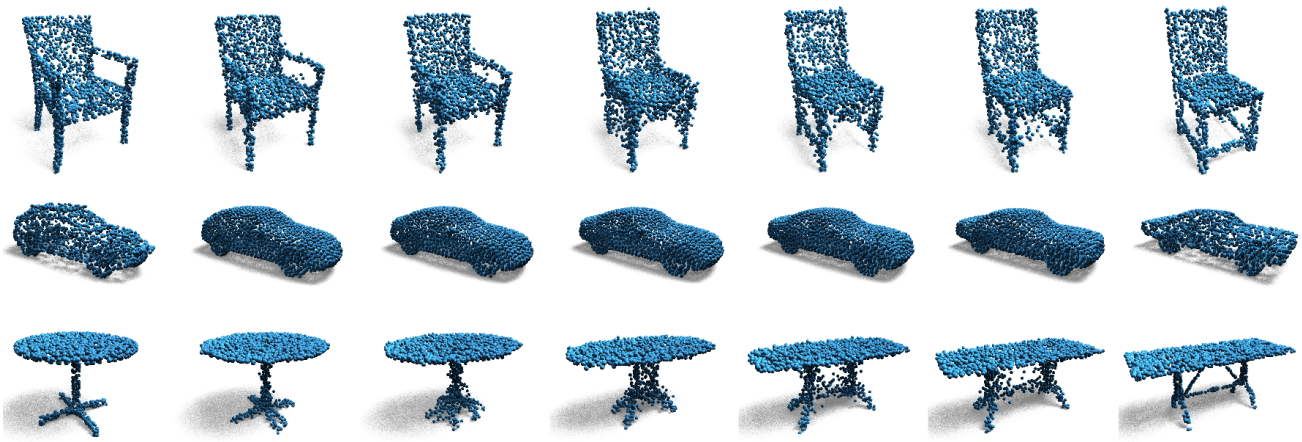

Figure 4. Interpolations between the test set objects obtained by our single-class AAE models. Leftmost and rightmost samples present ground truth objects. Images in between are the result of generating images from linear interpolation between our latent space encodings of the side images.

For training we use Adam [8] optimization algorithm with the following values of hyperparameters: learning rate equal to 0.0001 and $\beta 1$ value equal to 0.5 and $\beta 2$ value equal to 0.999. In all experiments we use scaling parameter λ that indicates the impact of reconstruction error in aggregated loss is equal 0.05.

5.3. Experimental setup

For all of the experiments we use *ShapeNet* dataset [4] transformed to the 3×2048 point cloud representation following the methodology provided in [1]. Unless otherwise stated, we train models with point clouds from a single object class and work with train/validation/test sets of an $85\% - 5\% - 10\%$ split. When reporting JSD measurements we use a 28^3 regular voxel grid to compute the statistics.

5.4. Models used for evaluation

The following reference models introduced in [1] are considered in experiments:

- **Autoencoder (AE)**. Simple architecture of autoencoder that converts input point cloud to the bottleneck representation z with encoder. The model is used for representing 3D points in latent space without any additional mechanisms that can be used to sample artificial 3D objects. Two approaches to train AE are considered: 1) with Chamfer (**AE-CD**) or 2) Earth-Mover (**AE-EMD**) distance as a way to calculate reconstruction error.
- **Raw point cloud GAN (r-GAN)**. Basic architecture of GAN, that learns to generate point clouds directly from the sampled latent vector.
- **Latent-space (Wasserstein) GAN (l-(W)GAN)**. Extended version of GAN trained in stacked mode on latent space z with and without an application of Wasserstein criterion [2].
- **Gaussian mixture model (GMM)**. Gaussian Mixture of Models fitted on the latent space of an encoder in the stacked mode.

Finally, we also evaluate the models proposed in this work as well as their variations:

- **Variational Autoencoder (3dVAE)**. Autoencoder with EMD reconstruction error and $KL(\cdot\|\cdot)$ regularizer.
- **Adversarial Autoencoder**. Adversarial autoencoder introduced in Section 4 that make use of prior $p(z)$ to learn the distribution directly on latent space z. Various types of priors are considered in our experiments: normal distribution (**3dAAE**), mixture of Gaussians (**3dAAE-G**).
- **Categorical Adversarial Autoencoder (3dAAE-C)**. AAE model with and additional category output returned by encoder for clustering purposes (see Section 5.9).

5.5. Reconstruction capabilities

In this experiment we evaluate the reconstruction capabilities of the proposed autoencoders using unseen test examples. We confront the reconstruction results obtained by AE model with our approaches to examine the influence of prior regularization on reconstruction quality. In Table 1 we report the MMD-CD and MMD-EMD between reconstructed point clouds and their corresponding ground-truth in the test dataset of the chair object class. It can be observed, that 3dVAE model does not suffer from overregularization problem raised in [1]. Both reconstruction measures are on the comparable level or even slightly lower for

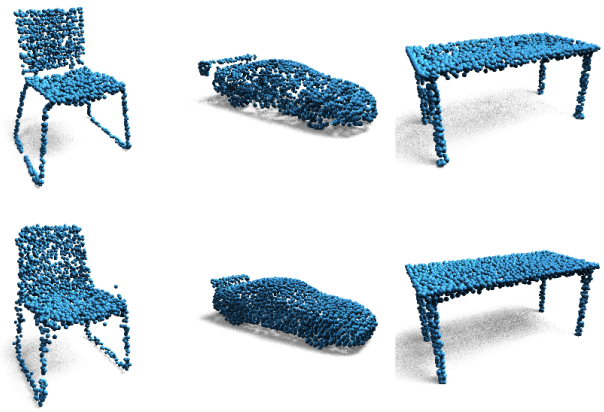

Figure 5. Reconstructions of objects from the test set obtained by our single-class 3dAAE models. Bottom row presents reconstructions (encoding and decoding) of corresponding ground truth objects from the top row.

Method	MMD-CD	MMD-EMD
AE [1]	0.0013	0.052
3dVAE	0.0010	0.052
3dAAE	0.0009	0.052
3dAAE-G	**0.0008**	**0.051**

Table 1. Reconstruction capabilities of the models captured by MMD. Measurements for reconstructions on the test split for the considered models trained with EMD loss and training data of the chair class.

3dAAE-G. For qualitative analysis (see Figure 5) we use our 3dAAEs to encode unseen samples from the test split (bottom row) and then decode and compare them visually to the input (top row).

5.6. Generative capabilities

We present the evaluation of our models, that are involved in sampling procedure directly on z latent space and compare them with the generative models trained in stacked mode basing on the latent representation of AE.

For evaluation purposes we use chair category and the following five measures: JSD, MMD-CD, MMD-EMD, COV-CD, COV-EMD. For each of the considered models we select the best one according to the JSD measure on validation set. To reduce the sampling bias of these measurements each generator produces a set of synthetic samples that is 3x the population of the comparative set (test or validation) and repeat the process 3 times and report the averages.

In Table 2 we report the generative result for memorization baseline model, 5 generative models introduced in [1] and three our approaches: 3dVAE, 3dAAE and 3dAAE-G. A baseline model memorizes a random subset of the training data of the same size as the other generated sets. For 3dAAE-G model we fix the number of Gaussian equal 32 with a different mean values and fixed diagonal covariance matrices.

It can be observed, that all of our approaches achieved the best results in terms of JSD measure. Practically, it means that our models are capable to learn better global statistics for generated point locations than reference solutions. We also noticed the slight improvement on MMD criteria for both of the considered distances. In terms of coverage criteria the results are comparable to the results obtained by the best GAN model and GMM approach.

In Table 3 we present an additional results on four categories: car, rifle, sofa and table. For further evaluation we use MMD-EMD and COV-EMD metrics. 3dAAE-G model achieved the best results considering MMD-EMD criterion for each of the datasets and the highest value for three of them. Practically, it means that the assumed Gaussian model with diagonal covariance matrix is sufficient to represent the data in latent space and training GMM in stacked mode is unnecessary when adversarial training with fixed prior is performed. In order to present qualitative result we provide synthetic samples generated by the model in Figure 1.

Method	JSD $\times 10^{-1}$	MMD-CD $\times 10^{-1}$	MMD-EMD $\times 10^{-1}$	COV-EMD	COV-CD
A[1]	0.17	0.018	0.630	78.6	79.4
B[1]	1.76	0.020	1.230	19.0	52.3
C[1]	0.48	0.020	0.790	32.2	59.4
D[1]	0.30	0.023	0.690	19.0	52.3
E[1]	0.22	0.019	0.660	66.9	67.6
F[1]	0.20	0.018	0.650	67.4	68.9
3dVAE	0.18	0.017	0.639	65.5	65.9
3dAAE	0.14	0.017	**0.622**	67.0	67.3
3dAAE-G	**0.14**	**0.017**	0.643	**68.7**	**69.6**

Table 2. Evaluating generative capabilities on the test split of the chair dataset on epochs/models selected via minimal JSD on the validation-split. We report A) sampling-based memorization baseline and the following reference methods from [1]: B) r-GAN, C) l-GAN (AE-CD), D) l-GAN (AE-EMD), E) l-WGAN (AE-EMD), F) GMM (AE-EMD). The last three rows refer to our approaches: 3dVAE, 3dAAE and 3dAAE-G.

5.7. Latent space arithmetic

One of the most important characteristic of well-trained latent representations is its ability to generate good-looking samples based on embeddings created by performing interpolation or simple linear algebra. It shows, that model is able to learn distribution of the data without excessive under- or overfitting [17].

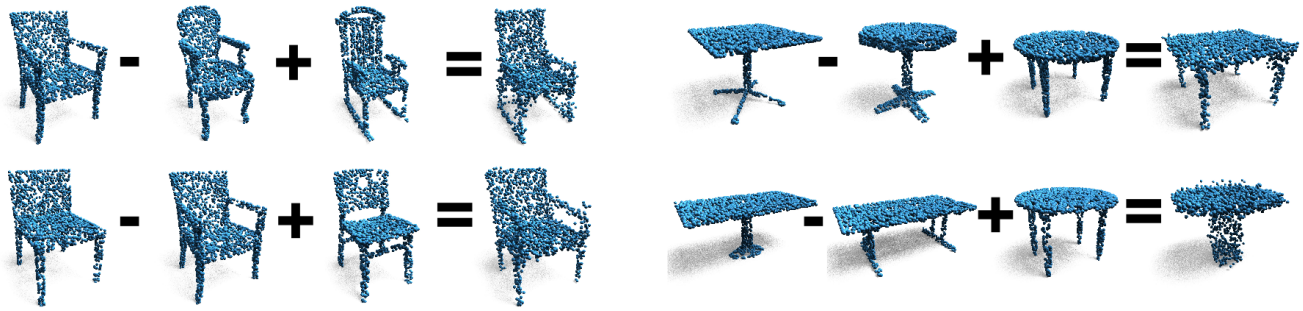

Figure 6. Modifying point clouds by performing additive algebra on our latent space encodings by our single-class AAE modes. Top-left sequence: adding rockers to a chair. Bottom-left: adding armrests to a chair. Top-right: changing table legs from one in the center to four in the corners. Bottom-right: changing table top from rectangle to circle-shaped.

Class	MMD-EMD			COV-EMD		
	PA	A	AG	PA	A	AG
car	0.041	0.040	**0.039**	65.3	66.2	**67.6**
rifle	0.045	0.045	**0.043**	74.8	72.4	**75.4**
sofa	0.055	0.056	**0.053**	**66.6**	63.5	65.7
table	**0.061**	0.062	**0.061**	71.1	71.3	**73.0**

Table 3. **COV-EMD** and **MMD-EMD** metrics on the test split of the car, rifle, sofa and table datasets on epochs/models selected via minimal JSD on the validation-split. We report the results for our 3dAAE (denoted as A) and 3dAAE-G (denoted by AG) models compared to the reference GMM model (denoted as PA) reported in [1].

Method	Numeric	Binary
AE	0.829	0.787
3dAAE	0.807	0.768
3dAAE-Bernoulli	0.913	0.892
3dAAE-Beta	**0.939**	**0.921**

Table 4. Retrieval results on *ShapeNet* dataset with 5 categories: car, rifle, sofa and table. We report results for numerical and binary features.

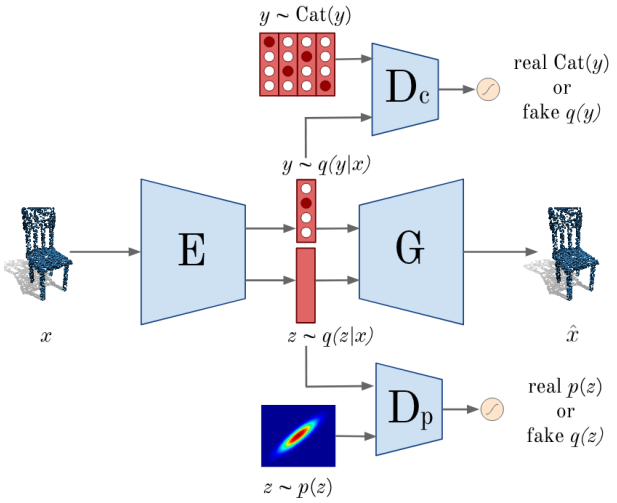

Figure 7. 3dAAE-C model architecture that extends 3dAAE with an additional decoder Dc that enable to learn an additional categorical representation y. The encoder is involved in two adversarial trainings to learn both categorical (y) and numerical (z) representations.

In Figure 4 we show that interpolation technique in the latent space produces smooth transitions between two distinct types of the same-class objects. It is worth mentioning that our model is able to change multiple characteristics of objects at once, *e.g.* shape and legs of the table.

Figure 6 presents model ability to learn meaningful encodings that allows to perform addition and subtraction in latent space in order to modify existing point clouds. In this examples, we were able to focus on specific feature that we want to add to our initial point cloud, while leaving other characteristics unchanged. Moreover, the operations were done by using only one sample for each part of the transformation.

5.8. Learning binary embedding with AAE models

In this section we present the results for learning binary feature embeddings. We make use of the benefit of using arbitrary selected prior distribution during the adversarial training to learn informative and diverse binary features. To achieve the stated goal we propose to sample z codes from Beta distribution that accumulates probability mass around 0 or 1 (by selecting low values of α and β hyperparameters) and sample the binary codes directly from Bernoulli distribution.

We consider 4 different approaches in our work: AE, 3dAAE, and two additional approaches: **3dAAE-Beta** — adversarial autoencoder trained with $Beta(\alpha = 0.01, \beta = 0.01)$ distribution of $p(z)$; **3dAAE-Bernoulli** — adversarial autoencoder trained with multivariate Bernoulli $Ber(p = 0.5)$ distribution of $p(z)$.

We examine the quality of the approaches for retrieval task taking under consideration mean average precision

(mAP) as quality criterion. For each of the considered models we select the best model according to the mAP value on validation set.

The results obtained on test set are reported in Table 4. We present the results both for numerical and binarized z values. The binarization is performed simply by taking threshold value equal 0 for 3dAAE and 0.5 for 3dAAE-Beta and 3dAAE-Bernoulli.

It can be observed, that models that force the diversity of coding space (3dAAE-Beta, 3dAAE-Bernoulli) achieved significantly better retrieval results than the models without these mechanisms (AE, 3dAAE).

5.9. Clustering 3D point clouds

Figure 8. Selected examples from test set clustered with adversarial autoencoder with an additional categorical units.

In this subsection we introduce the extension of adversarial model for 3D point clouds (**3dAAE-C**) inspired by [12], that incorporates an additional one-hot coding unit $\mathbf{y}$. This unit can be interpreted as a indicator of an abstract subclass for the 3D objects delivered on the input of the encoder.

The architecture for this model is provided in Figure 7 and extends the existing model (see Figure 2) with an additional discriminator D_c. The role of the discriminator is to distinguish between noise samples generated from categorical distribution and one-hot codes $\mathbf{y}$ provided by the encoder. During the training procedure the encoder is incorporated in additional task in which it tries to full the discriminator D_c to create samples from the categorical distribution. As a consequence, encoder is setting 1 value for the characteristic subcategories of the objects in the coding space z.

In Figure 8 we present the qualitative clustering results for *chair* test data. We trained the categorical adversarial autoencoder on chair dataset. We set the number of potential clusters to be extracted by our model to be equal 32. We selected 4 most dominant clusters and presented 6 randomly selected representatives for each of the them. It can be observed, that among the detected clusters we can observe the subgroups of chairs with characteristic features and shapes.

6. Conclusions

In this work we proposed an adversarial autoencoder for generating 3D point clouds. Contrary to the previous approaches [1] our model is an end-to-end 3D point cloud generative model that is capable to represent and sample artificial data from the same latent space. The possibility of use various priors in adversarial framework makes model useful not only for representation and generative purposes, but also in learning binary encoding and discovering hidden categories.

To show numerous capabilities of our model we provide various of experiments in terms of 3D points reconstruction, generation, retrieval and clustering. The results of the provided experiments confirm the good quality of the model in the mentioned tasks competitive for existing state-of-the-art solutions.

References

[1] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas. Learning representations and generative models for 3d point clouds. 2018. 1, 2, 4, 5, 6, 7, 8

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 5

[3] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 1

[4] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 5

[5] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, volume 2, page 6, 2017. 1, 2

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1, 2

[7] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017. 3

[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[9] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1, 2

[10] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951. 4

[11] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, volume 2, page 4, 2017. 3

[12] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow. Adversarial autoencoders. In *International Conference on Learning Representations*, 2016. 3, 8

[13] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017. 1, 2, 3

[14] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 1

[15] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:99–121, 2000. 2

[16] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler. Semantic visual localization. *CoRR*, abs/1712.05773, 2017. 2

[17] F. P. Tasse and N. Dodgson. Shape2vec: semantic-based descriptors for 3d shapes, sketches and images. *ACM Transactions on Graphics (TOG)*, 35(6):208, 2016. 6

[18] J. Tomczak and M. Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223, 2018. 1

[19] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016. 1, 2

[20] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan. Supervised hashing for image retrieval via image representation learning. In *AAAI*, volume 1, page 2, 2014. 1

[21] M. Zieba, P. Semberecki, T. El-Gaaly, and T. Trzcinski. Bingan: Learning compact binary descriptors with a regularized gan. *arXiv preprint arXiv:1806.06778*, 2018. 1

[22] M. Zieba and L. Wang. Training triplet networks with gan. *arXiv preprint arXiv:1704.02227*, 2017. 1