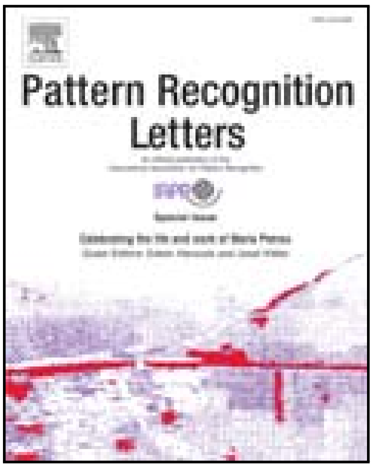
Journal Pre-proof



Transcoding across 3D Shape Representations for Unsupervised Learning of 3D Shape Feature

Takahiko Furuya , Ryutarou Ohbuchi

PII: S0167-8655(20)30258-0
DOI: https://doi.org/10.1016/j.patrec.2020.07.012
Reference: PATREC 7961

To appear in: *Pattern Recognition Letters*

Received date: 28 August 2019
Revised date: 5 June 2020
Accepted date: 11 July 2020

Please cite this article as: Takahiko Furuya , Ryutarou Ohbuchi , Transcoding across 3D Shape Representations for Unsupervised Learning of 3D Shape Feature, *Pattern Recognition Letters* (2020), doi: https://doi.org/10.1016/j.patrec.2020.07.012

Highlights

- Proposing "transcoding 3D shape representations" for unsupervised feature learning.
- Implementing the approach as a DNN called Shape Auto-Transcoder (SAT).
- Evaluating SAT under scenarios of retrieval and classification of 3D shapes.

Transcoding across 3D Shape Representations for Unsupervised Learning of 3D Shape Feature

Takahiko Furuya[a, *] and Ryutarou Ohbuchi[a]

[a] *University of Yamanashi, 4-3-11 Takeda, Kofu-shi, Yamanashi-ken, 400-8511, Japan*

ABSTRACT

Unsupervised learning of 3D shape feature is a challenging yet important problem for organizing a large collection of 3D shape models that do not have annotations. Recently proposed neural network-based approaches attempt to learn meaningful 3D shape feature by autoencoding a single 3D shape representation such as voxel, 3D point set, or multiview 2D images. However, using single shape representation isn't sufficient in training an effective 3D shape feature extractor, as none of existing shape representation can fully describe geometry of 3D shapes by itself. In this paper, we propose to use *transcoding across multiple 3D shape representations* as the unsupervised method to obtain expressive 3D shape feature. A neural network called Shape Auto-Transcoder (SAT) learns to extract 3D shape features via cross-prediction of multiple heterogeneous 3D shape representations. Architecture and training objective of SAT are carefully designed to obtain effective feature embedding. Experimental evaluation using 3D model retrieval and 3D model classification scenarios demonstrates high accuracy as well as compactness of the proposed 3D shape feature. The code of SAT is available at https://github.com/takahikof/ShapeAutoTranscoder

Keywords: unsupervised feature learning; 3D shape retrieval and classification; deep learning; autoencoder

1. Introduction

3D shape model has seen widespread applications in such areas as mechanical design, computer graphics, autonomous robotics, and medical diagnosis. To effectively organize the 3D models, technologies for shape similarity-based indexing, clustering, or retrieval of 3D shapes have become necessary. Typically, shape similarities among 3D models are computed by using 3D shape features, or 3D shape descriptors, that characterize shapes of 3D models as multidimensional vectors. Accurate yet compact 3D shape feature is essential for effective and efficient comparison among a large number of 3D shapes.

Recently, Deep Neural Networks (DNNs) that directly or indirectly process 3D shapes have been proposed [1][2][3][4]. The DNNs take as their input 3D shapes defined by one of diverse shape representations, for example, 3D point set [1], voxels [2], polygonal mesh [3], or multi-view 2D images [4]. Most of the DNNs learn the 3D shape features in a supervised manner by associating the 3D shapes with semantic labels. However, the existing labeled 3D model datasets are small in size and lack diversity due to cost of annotation. Consequently, an effective supervised learning of DNN for 3D shape feature is often not practicable.

Insufficiency of labeled 3D shapes has prompted studies on unsupervised learning of 3D shape feature. Unsupervised approach potentially allows learning of 3D shape features from an unlabeled yet large collection of 3D shapes. However, unsupervised learning is more challenging than supervised learning. To overcome the challenge, surrogate tasks are employed to train DNNs using the unlabeled 3D shapes. One of the most representative surrogate tasks is autoencoding, or self-reconstruction. Each of the recently proposed 3D shape autoencoders processes a single 3D shape representation. For example, Sharma et al. [5] proposed the volumetric autoencoder for processing 3D shapes represented as voxels. Yang et al. [6] proposed the autoencoder tailored to 3D point sets. Zhu et al. [7] and Leng et al. [8] devised 3D shape matching methods using the autoencoder that accepts 3D shapes rendered as a set of 2D images. These studies showed that the 3D shape autoencoders could produce 3D shape features more accurate than conventional handcrafted 3D shape features. However, accuracy of the 3D shape features learned by using the 3D shape autoencoders is still insufficient for many practical applications.

We suspect that insufficient accuracy of the autoencoder-based 3D shape features stems from limited expressive power of input 3D shape representation. Any one of shape representation, e.g., 3D point set, voxels, or multiview 2D images, would have difficulty in fully describing 3D geometry of the 3D shapes. For example, 3D point set is unstructured data. Thus, it is unable to explicitly represent surfaces of the 3D shapes and connections among their parts. Voxel representation usually has low resolution (e.g., 32^3) to contain computational cost. Such coarse voxel grids fail to represent shape details. Multiview rendering of 3D shape generally does not capture internal structure of the 3D shape. Using a single 3D shape representation for training limits effective learning of 3D shape feature.

* Corresponding author. E-mail: takahikof@yamanashi.ac.jp

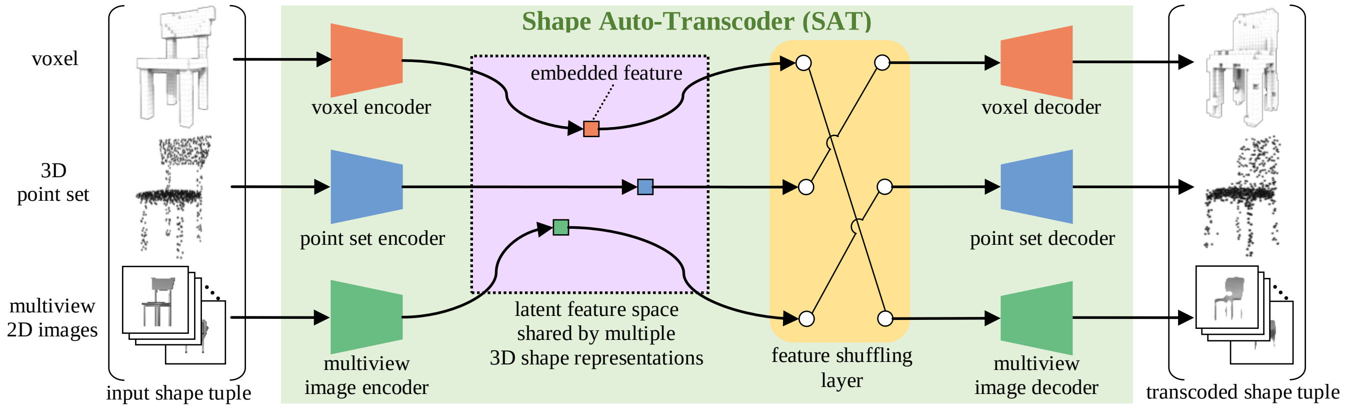


Fig. 1. This paper proposes a novel approach to unsupervised learning of 3D shape feature via *transcoding*. Using the approach, Shape Auto-Transcoder (SAT) learns 3D shape feature by transcoding, or cross-converting, across multiple 3D shape representations.

Our goal here is to obtain expressive and compact 3D shape features under the framework of unsupervised learning. To achieve this goal, we propose a simple yet effective approach to unsupervised learning of 3D shape feature. Our key idea is to take advantage of information in multiple 3D shape representations by means of "*transcoding*". Here, the term "transcoding" means cross-conversion between pairs of 3D shape representations. The idea substantially differs from the conventional approaches that autoencode just one 3D shape representation. Fig. 1 illustrates proposed *Shape Auto-Transcoder* (*SAT*) DNN. SAT synergistically learns expressive 3D shape feature via transcoding across multiple shape representations, each of which should capture different aspects of 3D geometry of 3D shapes. During the training, encoder DNNs embed multiple input shape representations of a 3D shape into a low-dimensional latent feature space shared among the input representations. After feature embedding, the latent features are "shuffled" and then fed into decoder DNNs to reconstruct shape representations possibly different from their respective input representations. Effective training of SAT is achieved by using objective functions tailored for unsupervised feature embedding. In addition to shape reconstruction loss, *feature agglomeration* loss is used to consistently align distributions of embedded features across shape representations.

After the training, the encoders of SAT are used as feature extractors. A properly trained SAT should embed a 3D shape at a point, or, nearly a point, in the common feature space regardless of shape representation used. The feature can then be used to compare 3D shapes having same (e.g., voxel) or different (e.g., voxel and point set) shape representations.

We empirically evaluate the 3D shape feature learned by SAT under scenarios of 3D shape retrieval and 3D shape classification. Experimental results demonstrate that SAT produces 3D shape feature more accurate than the existing unsupervised approaches that learn from a single 3D shape representation.

Contribution of this paper can be summarized as follows.

- Proposing an approach for unsupervised learning of 3D shape feature via transcoding across multiple 3D shape representations.
- Implementing the approach as a DNN called Shape Auto-Transcoder (SAT). SAT is trained by using objective functions tailored to it.
- Empirically evaluating the SAT by using scenarios of 3D shape retrieval and 3D shape classification.

The rest of the paper is organized as follows. Related work is reviewed in Section 2 and our proposed approach is described in Section 3. Section 4 reports experimental results. Finally, conclusion and future work are discussed in Section 5.

2. Related work

2.1. Unsupervised learning of 2D image feature

Unsupervised feature learning has become a hot research topic especially in the field of 2D image analysis. Much effort has been made to obtain meaningful visual features from a large collection of unlabeled 2D images. Recently proposed approaches often involve unsupervised learning of 2D-Convolutional Neural Network (2D-CNN). Since labels are unavailable, the 2D-CNNs are trained by self-supervision, where supervisory signals are created from unlabeled input 2D images.

Autoencoder (AE) [9] is a powerful framework for unsupervised feature learning. AE comprises a pair of an encoder DNN and a decoder DNN. The encoder, which acts as a feature extractor, embeds input data to a low-dimensional latent feature space. The decoder attempts to reconstruct the input from the latent feature. Learning via self-reconstruction is stable. That is, training of AEs usually converges to one of suboptimal solutions that can reconstruct diverse training data.

Generative Adversarial Network (GAN) [10][11] learns data generation and feature extraction in an unsupervised manner. A GAN consists of a generator DNN and a discriminator DNN. The generator attempts to produce realistic data. The discriminator tries to classify its input either as real data or fake data, the latter of which is produced by the generator. Visual features derived from the trained discriminator typically outperform features learned by using AEs [10][11]. However, training of GAN is often unstable since maintaining balance between the generator and the discriminator is difficult [12].

In addition to 2D image, AE and GAN are also applicable to 3D shape as described in Section 2.2. Other unsupervised approaches for visual feature learning include, for example, predicting context of image patches [13], colorizing images [14], classifying pseudo classes created from unlabeled images [15][16]. However, these approaches cannot be directly applied to 3D shape since their DNN architectures and training objectives are designed specifically for 2D image.

2.2. Unsupervised learning of 3D shape feature

Traditional 3D shape matching algorithms employed handcrafted 3D shape features [17]. These algorithms are designed for feature extraction from such 3D shape representations as manifold mesh [18][19], 3D point set [20][21], voxels [22][23], and multi-view 2D images [24][25]. The handcrafted features work reasonably well in many applications that require 3D shape matching. However, the handcrafted features are not data driven, that is, not optimized for a specific dataset to be analyzed.

Data-driven approaches thus have been proposed to obtain expressive 3D shape feature. Sharma et al. [5], Brock et al. [26], and Wang et al. [45] proposed volumetric autoencoders that accept 3D shapes represented as voxels. Wu et al. [27] devised a GAN for voxels called 3D-GAN. These voxel-based DNNs capture hierarchical geometric feature of 3D shape by using 3D-Convolutional Neural Network (3D-CNN). Yang et al. [6] proposed FoldingNet, which is an autoencoder for 3D point set. The encoder part of FoldingNet first computes per-point features by using a neighborhood graph structure of 3D points. The per-point features are then aggregated, by max pooling, to a latent feature per 3D shape that is invariant against permutation of input 3D points. The decoder of FoldingNet reconstructs shape of the input point set by deforming a 2D regular grid. Achlioptas et al. [28] proposed the GAN for 3D point set called Latent GAN. To stabilize its training, Latent GAN generates and classifies latent shape features learned by using AE, rather than raw 3D point sets. VIP-GAN by Han et al. [46] is a GAN tailored to multiview 2D images. Zhu et al. [7] and Leng et al. [8] utilized 2D-CNNs to learn visual features of 3D shapes which are represented as sets of multiview 2D images. Hausdorff distance is used to compare the sets of visual features extracted from the 2D views.

Those DNN-based unsupervised approaches above are able to learn 3D shape features more powerful than the pre-DNN, handcrafted ones. We argue, however, that accuracy of the features using DNN-based unsupervised approach can be improved further. We postulate that accuracy of the current DNN-based approaches is limited because they use single shape representation as their source of information. As we mentioned in Section 1, by recruiting more than one shape representations with their mutually complementary geometrical information, we should be able to learn more expressive feature.

2.3. Cross-modal autoencoder

Cross-modal autoencoders, or cross-modal AEs, have been widely studied in the field of cross-modal information retrieval [29]. Cross-modal AEs enable similarity comparison among data from different domains (e.g., texts and 2D images) by learning their common latent feature space. Similar to the standard, unimodal AEs described in Section 2.1, cross-modal AEs can be trained in an unsupervised fashion via reconstructing multimodal input data. Earlier cross-modal AEs [30][31][32] have DNN architectures and training objectives tailored to vector representation extracted from multimodal data.

Recently, aiming at effective unsupervised learning of visual feature, cross-modal AEs that directly process raw 2D images or 2.5D images have been proposed. Split-brain AE proposed by Zhang et al. [33] splits a 2D image into two different modalities, i.e., luminance and color, and cross-predicts across these modalities by using convolutional AEs. Visual feature learned by the split-brain AE significantly outperforms feature obtained from the standard AEs. Kuga et al. [34] proposed cross-prediction of color and depth of RGB-D images to obtain image features useful for semantic segmentation of RGB-D images.

Our proposed SAT can be viewed as a variant of cross modal AEs since SAT cross-predicts across different 3D shape representations. We stress, however, that network architecture and training objectives for SAT are highly customized to 3D shapes defined by using multiple shape representations. To our best knowledge, our study is the first attempt that introduces transcoding of 3D shape representations to the problem of unsupervised learning of 3D shape feature.

3. Proposed algorithm

3.1. Overview of Shape Auto-Transcoder

We propose a simple yet effective approach to unsupervised learning of 3D shape feature. The DNN called Shape Auto-Transcoder (SAT), illustrated in Fig. 1, learns 3D shape feature via transcoding across multiple heterogeneous 3D shape representations. Assuming N 3D shape representations to train SAT, there are N encoders, a shared latent feature space, and N decoders. There is also a layer that shuffles connection between latent feature and decoder. N is 3 in this paper, for we chose three shape representations, voxels, 3D point set, and multiview 2D images to train SAT. Note that more 3D shape representation (e.g., manifold mesh) can be added to SAT if AEs for the shape representation is available. SAT is trained by transcoding across all the N^2 combinations of input/output shape representations. In contrast, existing 3D shape AEs learn by encoding and decoding shapes in a same shape representation. By using information available in multiple 3D shape representations, SAT is expected to capture more expressive 3D shape feature than unimodal AEs.

SAT uses a set of N-tuple for its training. Each tuple, called "shape tuple", consists of N shape representations of a 3D shape (see Fig. 1). Using a set of polygonal 3D CAD models as the source, a set of shape tuples is generated for training the SAT. To transcode, given an input shape tuple of a 3D model S, N encoders of the SAT embed the N representations of S into a latent feature vector F in a space shared across the N representations. Then, N decoders reconstruct F into N shape representations of S. In doing so, a shape S in representation A may be reconstructed into any one of N shape representations. That is, when a representation A of a shape S is reconstructed into a representation B of the shape S, A may be different from B. This scrambling of correspondence is done by the feature shuffling layer placed between latent space and the decoder.

Custom training objective function is devised to train SAT. Similar to standard AE, we employ shape reconstruction losses for each of N shape representations. In addition, we propose to use *feature agglomeration loss* that prompts distribution of latent features to be consistently aligned among different shape representations. After the training, one of encoders of SAT is used to extract 3D shape features. If more than one representations of a 3D shape are available, multiple encoders for the available shape representations may be combined for a more expressive feature. To do so, for example, multiple feature vectors of a 3D shape produced by multiple encoders may be concatenated together to form a feature vector for the 3D shape.

3.2. Architecture of SAT

3.2.1. Shape transcoding

Transcoding of 3D shape representations is achieved by using multiple 3D shape AEs combined with a feature shuffling layer placed between the encoders and the decoders. Suppose that $\mathrm{T} = (\mathbf{S}_{vx}, \mathbf{S}_{ps}, \mathbf{S}_{mv})$ is an input shape tuple where $\mathbf{S}_{vx}$, $\mathbf{S}_{ps}$, and $\mathbf{S}_{mv}$ corresponds, respectively, to 3D shapes represented as voxels, point set, and multiview images. The encoder part of 3D shape AEs transforms its corresponding input 3D shape representations to the embedded features. The embedded features, denoted by vectors $\mathbf{f}_{vx}$, $\mathbf{f}_{ps}$, and $\mathbf{f}_{mv}$, have the same number of dimensions n_e

(e.g., n_e=128) and are normalized by their Euclidean norms. As N = 3, the feature vectors form a matrix **F** having size of 3×n_e.

The feature shuffling layer randomly alters ordering of the three embedded feature vectors. As shown in Eq. 1, feature shuffling can be computed by multiplying the permutation matrix **P** and the feature matrix **F**.

$$\mathbf{F}_{\mathrm{shuffled}} = \mathbf{PF} = \mathbf{P}\begin{pmatrix}\mathbf{f}_{\mathrm{vx}} \\ \mathbf{f}_{\mathrm{ps}} \\ \mathbf{f}_{\mathrm{mv}}\end{pmatrix} \qquad (1)$$

The permutation matrix **P** is created by randomly permutating rows (or columns) of 3×3 identity matrix. Note that, during training, **P** is recomputed for every input shape tuple. A sufficiently large training iterations thus covers all the possible combinations (i.e., 3^2=9 for N = 3) of input-output shape representation pairs for transcoding every shape tuple.

After shuffling correspondences among input and output shape representations, the embedded features are fed into the decoders to generate transcoded 3D shapes. The embedded feature in the first row of $\mathbf{F}_{\mathrm{shuffled}}$ is passed on to the voxel decoder. Similarly, the second row (or the third row) of $\mathbf{F}_{\mathrm{shuffled}}$ is fed into the point set decoder (or the multiview image decoder). Consequently, the decoders produce a tuple of three transcoded 3D shapes $\hat{T} = (\hat{\mathbf{S}}_{\mathrm{vx}}, \hat{\mathbf{S}}_{\mathrm{ps}}, \hat{\mathbf{S}}_{\mathrm{mv}})$.

3.2.2. Shape encoders

Table 1 summarizes architectures of the encoders for processing the three 3D shape representations.

Voxel encoder: The input shape $\mathbf{S}_{\mathrm{vx}}$ is represented as 3D occupancy grids whose resolution is 32^3. Each grid cell has value 1 if it overlaps with any polygon of the 3D CAD model, or 0 otherwise. We construct the 3D-CNN with residual connection similar to [26]. The input 3D grids are processed, in sequence, by a 3D convolution ("3D conv.") layer, six 3D residual ("3D res.") blocks, and two fully-connected ("FC") layers to produce the n_e-dimensional latent feature $\mathbf{f}_{\mathrm{vx}}$. In Table 1, for example, "3D conv. (3^3, 16, 1)" indicates 3D convolution using 16 filters of size 3^3 and stride 1. The "3D res. (3^3, 32, 1)" convolves an input feature map twice using filters specified in the parentheses and adds the convolved feature map to the input feature map. "FC (512)" denotes an FC layer having 512 neurons.

Point set encoder: Each point set $\mathbf{S}_{\mathrm{ps}}$ consists of 1,024 3D points randomly and uniformly sampled on surfaces of 3D CAD model. $\mathbf{S}_{\mathrm{ps}}$ is normalized to be zero-mean and enclosed in a unit sphere. We adopt PointNet [1] as the encoder DNN. The first five FC layers extract a set of per-point features from the input point set. The per-point features are then aggregated, by max pooling, to a single feature vector per 3D point set. The pooled feature is processed further by the subsequent two FC layers to produce the embedded feature $\mathbf{f}_{\mathrm{ps}}$ having n_e dimensions.

Multiview image encoder: The input 3D shape $\mathbf{S}_{\mathrm{mv}}$ is represented as a set of 12 grayscale 2D images of size 64^2 each. The 3D CAD model is rendered from cameras located at 12 vertices of a regular icosahedron that encloses the 3D model. We base our multiview image encoder on ResNet [35]. Each image is first encoded to a per-view visual feature by using a 2D convolution ("2D conv.") followed by eight 2D residual ("2D res.") blocks. Each 2D residual block convolves an input feature map twice, and the convolved feature map is then added to the input feature map. The set of 12 per-view features is aggregated by max-pooling. The pooled feature is embedded, via the subsequent two FC layers, as the n_e-dimensional latent feature $\mathbf{f}_{\mathrm{mv}}$.

For all the encoders described above, output of each hidden layer is normalized by batch normalization [36] and then activated by ReLU function [37]. No activation function is applied to the output from the last FC layer of encoders.

3.2.3. Shape decoders

Table 2 summarizes architectures of our 3D shape decoders. The voxel decoder transforms the embedded feature, which is selected at the feature shuffling layer, to the transcoded voxel representation $\hat{\mathbf{S}}_{\mathrm{vx}}$. The voxel decoder has one FC layer followed by five 3D deconvolution ("3D deconv.") layers. Similarly, the multiview image decoder maps the embedded feature to the set of 12 2D images $\hat{\mathbf{S}}_{\mathrm{mv}}$ by using one FC layer and five 2D deconvolution ("2D deconv.") layers. The point set decoder generates a set of 1,024 3D points $\hat{\mathbf{S}}_{\mathrm{ps}}$ via three FC layers.

Batch normalization followed by ReLU activation is applied to all the hidden layers of the decoders. To constrain values of decoder outputs, the values of transcoded voxels and multiview image pixels are activated by sigmoid function, while the coordinate value of transcoded set of points is activated by hyperbolic tangent function.

Table 1
Encoder architectures of SAT.

voxel encoder		point set encoder		multiview image encoder	
layers	output size	layers	output size	layers	output size
—	32^3×1	—	1024×3	—	12×64^2×1
3D conv. (3^3, 16, 1)	32^3×16	FC (64)	1024×64	2D conv. (3^2, 32, 1)	12×64^2×32
maxpool	16^3×16	FC (64)	1024×64	maxpool	12×32^2×32
3D res. (3^3, 32, 1)	16^3×32	FC (64)	1024×64	2D res. (3^2, 64, 1)	12×32^2×64
3D res. (3^3, 32, 1)	16^3×32	FC (128)	1024×128	2D res. (3^2, 64, 1)	12×32^2×64
maxpool	8^3×32	FC (1024)	1024×1024	maxpool	12×16^2×64
3D res. (3^3, 64, 1)	8^3×64	maxpool	1×1024	2D res. (3^2, 128, 1)	12×16^2×128
3D res. (3^3, 64, 1)	8^3×64	FC (512)	512	2D res. (3^2, 128, 1)	12×16^2×128
maxpool	4^3×64	FC (n_e)	n_e	maxpool	12×8^2×128
3D res. (3^3, 128, 1)	4^3×128			2D res. (3^2, 128, 1)	12×8^2×128
3D res. (3^3, 128, 1)	4^3×128			2D res. (3^2, 128, 1)	12×8^2×128
maxpool	2^3×128			maxpool	12×4^2×128
FC (512)	512			2D res. (3^2, 256, 1)	12×4^2×256
FC (n_e)	n_e			2D res. (3^2, 256, 1)	12×4^2×256
				maxpool	1×4^2×256
				FC (512)	512
				FC (n_e)	n_e

Table 2
Decoder architectures of SAT.

voxel decoder		point set decoder		multiview image decoder	
layers	output size	layers	output size	layers	output size
—	n_e	—	n_e	—	n_e
FC (65536)	8^3×128	FC (512)	512	FC (98304)	12×4^2×512
3D deconv. (3^3, 64, 1)	8^3×64	FC (1,024)	1024	2D deconv. (3^2, 256, 2)	12×8^2×256
3D deconv. (3^3, 32, 2)	16^3×32	FC (3,072)	1024×3	2D deconv. (3^2, 128, 2)	12×16^2×128
3D deconv. (3^3, 16, 1)	16^3×16			2D deconv. (3^2, 64, 2)	12×32^2×64
3D deconv. (3^3, 8, 2)	32^3×8			2D deconv. (3^2, 32, 2)	12×64^2×32
3D deconv.	32^3×1			2D deconv.	12×64^2×1

$(3^3, 1, 1)$			$(3^2, 1, 1)$	

3.3. Training of SAT

3.3.1. Objective function

SAT is trained so that the loss function $L = L_{\text{vx}} + L_{\text{ps}} + L_{\text{mv}} + L_{\text{fa}}$ is minimized. Each term of L is described below.

Voxel transcoding loss $\boldsymbol{L}_{\textbf{vx}}$: We use weighted binary cross entropy [26] denoted by Eq. 2. In the equation, t_i and o_i indicate density of i-th voxel of ground truth shape $\mathbf{S}_{\text{vx}}$ and transcoded shape $\hat{\mathbf{S}}_{\text{vx}}$, respectively. γ controls penalty for false positives and false negatives. We use γ = 0.97 as suggested in [26].

$$L_{\text{vx}} = \sum_{i=1}^{32^3} -\gamma\ t_i \log\left(o_i\right) - \left(1-\gamma\right)\left(1-t_i\right)\log\left(1-o_i\right) \tag{2}$$

Point set transcoding loss $\boldsymbol{L}_{\textbf{ps}}$: We employ chamfer distance shown in Eq. 3 as set-to-set distance between ground truth point set $\mathbf{S}_{\text{ps}}$ and transcoded point set $\hat{\mathbf{S}}_{\text{ps}}$.

$$L_{\text{ps}} = \sum_{\mathbf{x}\in\mathbf{S}_{\text{ps}}} \min_{\mathbf{y}\in\hat{\mathbf{S}}_{\text{ps}}} \left\|\mathbf{x}-\mathbf{y}\right\|_2 + \sum_{\mathbf{y}\in\hat{\mathbf{S}}_{\text{ps}}} \min_{\mathbf{x}\in\mathbf{S}_{\text{ps}}} \left\|\mathbf{x}-\mathbf{y}\right\|_2 \tag{3}$$

Multiview image transcoding loss $\boldsymbol{L}_{\textbf{mv}}$: We use sum of L1 distances denoted by Eq. 4 as transcoding loss of multiview image representation. In Eq. 4, t_{ij} (or o_{ij}) indicates j-th pixel value of i-th image of ground truth multiview images $\mathbf{S}_{\text{mv}}$ (or transcoded multiview images $\hat{\mathbf{S}}_{\text{mv}}$).

$$L_{\text{mv}} = \sum_{i=1}^{12}\sum_{j=1}^{64^2} \left\|t_{ij} - o_{ij}\right\|_1 \tag{4}$$

Feature agglomeration loss $\boldsymbol{L}_{\textbf{fa}}$: In addition to the shape transcoding losses described above, we propose to use feature agglomeration loss defined by Eq. 5 for better feature embedding.

$$L_{\text{fa}} = \sum_{\mathbf{f}\in\left(\mathbf{f}_{\text{vx}},\mathbf{f}_{\text{ps}},\mathbf{f}_{\text{mv}}\right)} \max\left(0,\ \left\|\mathbf{f}-\boldsymbol{\mu}_{\text{pos}}\right\|_2 - \left\|\mathbf{f}-\boldsymbol{\mu}_{\text{neg}}\right\|_2 + 1\right) \tag{5}$$

The embedded features $\mathbf{f}_{\text{vx}}$, $\mathbf{f}_{\text{ps}}$, and $\mathbf{f}_{\text{mv}}$ extracted from an input shape tuple are pulled closer to their mean $\boldsymbol{\mu}_{\text{pos}}$ than the mean $\boldsymbol{\mu}_{\text{neg}}$ of features extracted from the shape tuples other than the input shape tuple. The shape tuples for computing $\boldsymbol{\mu}_{\text{neg}}$ are randomly selected from a mini-batch of input shape tuples. Minimizing L_{fa} forces the embedded feature distributions to align consistently among multiple 3D shape representations. An agglomerated feature is expected to embody rich information about geometry of a 3D shape since the feature could be decoded into any one of shape representations used for training.

We use Adam optimizer [38] with initial learning rate 0.001 to minimize the overall loss function L. Parameters of SAT are randomly initialized by using the algorithm by He et al. [39]. Each mini-batch contains eight shape 3-tuples. We iterate training of SAT for 200 epochs.

3.3.2. Data augmentation

A 3D feature is often need to be robust against translation, rotation, or noisy variation in coordinate values of 3D shape. We thus perform online data augmentation during SAT training. Voxel representation $\mathbf{S}_{\text{vx}}$ is randomly rotated, about its upright axis, by either of {0, 90, 180, 270} degrees. Then, $\mathbf{S}_{\text{vx}}$ is randomly shifted by [−2, +2] voxels along each axis. Every 3D point within $\mathbf{S}_{\text{ps}}$ is randomly and independently jittered by value sampled from a normal distribution $N(0, 0.01)$. $\mathbf{S}_{\text{ps}}$ is also randomly rotated in the same manner as voxel representation. For multiview image representation $\mathbf{S}_{\text{mv}}$, every rendered 2D image is randomly shifted by [−4, +4] pixels along each axis.

4. Experiments and results

4.1. Experimental setup

We evaluate efficacy of the 3D shape feature learned by SAT under 3D shape retrieval and 3D shape classification scenarios.

Datasets: We use both synthetic and realistic 3D shape data to evaluate generality of the proposed SAT. As synthetic 3D shape datasets, we use ModelNet10 (MN10) and ModelNet40 (MN40) [40]. MN10 consists of the training set of 3,991 3D CAD models and the test set of 908 3D CAD models classified into 10 object categories such as bathtub, chair, and table. MN40 contains 9,843 training 3D models and 2,468 3D models belonging to 40 categories such as airplane, plant, and car. As a realistic 3D shape dataset, we adopt ScanObjectNN dataset [43]. ScanObjectNN contains 3D shapes reconstructed from RGB-D scans. The 3D shapes reconstructed from scans are significantly different from the CAD models since the reconstructed shapes have holes and cracks on the surface of objects as well as background noise. In our experiments, we used the subset named OBJ_ONLY where each reconstructed shape is preprocessed to remove backgrounds such as wall or floor. We used 2,309 training 3D shapes and 581 testing 3D shapes classified into 15 indoor object categories such as chair, door, shelf, and sofa. Since the 3D shapes in ScanObjectNN are represented as colored 3D point sets, we converted them to polygonal 3D models by using the ball-pivoting algorithm [47] implemented in MeshLab [44]. We omitted color of the 3D points as the SAT as-is does not handle color. Shape tuples were then created from the converted polygonal 3D models and were used for training and evaluation of SAT.

To evaluate retrieval accuracy, SAT is first trained by using shape tuples created from 3D polygonal models in the training set. Note that, object category labels attached to the training 3D models are not used since our object is unsupervised (or, self-supervised) training of SAT. After the training, retrieval accuracy is calculated by using the 3D models in the test set. Mean Average Precision (MAP) [%] is used as the numerical measure of retrieval accuracy.

Classification accuracy is evaluated as follows. For the MN10 and MN40 datasets, we follow the evaluation protocol adopted by the previous studies [6][27][28] for fair comparison. That is, SAT is first trained by using a set of nearly 50,000 3D models contained in the ShapeNet Core 55 dataset [41]. By using the encoders of the trained SAT, 3D shape features are extracted from the 3D models of MN10 or MN40. A linear SVM is then trained by using the features extracted from the training 3D models. Finally, the features of testing 3D models are classified by the SVM. For the ScanObjectNN dataset, SAT is trained by using 3D models in the training set. After training SAT, a linear SVM is trained by using the features extracted from the training set and the SVM is used to classify the 3D shape features of the test set. Micro-averaged accuracy [%] is used for the evaluation.

Competitors: We compare the 3D shape feature learned by SAT against nine existing 3D shape features. They are two handcrafted features and seven features learned in unsupervised fashion. SPherical Harmonic descriptor (SPH) [22] and Light Field Descriptor (LFD) [24] are handcrafted features designed for voxel and multiview images respectively. Vconv-DAE [5], FoldingNet [6], Multiview Autoencoder (MVAE) [7], and CAE-ELM [45] are autoencoders for 3D shape, each of which accepts a single 3D shape representation. 3D-GAN [27], Latent-GAN

Table 3
Accuracy comparison with the existing algorithms on the synthetic ModelNet datasets. (V: voxel, P: point set, M: multiview images, *: our implementation)

algorithm	3D shape representation		feature dim.	retrieval accuracy (MAP)		classification accuracy	
	training	feature extraction		MN10	MN40	MN10	MN40
SPH	—	V	544	44.1	33.3	79.8	68.2
LFD	—	M	4,700	49.8	40.9	79.9	75.5
Vconv-DAE	V	V	6,192	—	—	80.5	75.5
Vconv-DAE*	V	V	6,192	65.2	49.2	90.9	87.3
CAE-ELM	V	V	—	—	—	91.4	84.4
3D-GAN	V	V	7,168	—	—	91.0	83.3
Latent-GAN	P	P	512	—	—	**95.3**	85.7
VIP-GAN	M	M	4,096	—	—	90.2	**92.2**
FoldingNet	P	P	512	—	—	94.4	88.4
FoldingNet*	P	P	512	67.3	52.4	91.2	86.6
MVAE*	M	M	30	50.1	31.2	—	—
SAT	V, P, M	V	64	71.9	58.7	90.5	86.3
SAT	V, P, M	P	64	71.8	58.7	91.1	85.9
SAT	V, P, M	M	64	70.3	60.5	91.0	86.5
SAT	V, P, M	V, P	128	72.5	59.7	91.9	87.2
SAT	V, P, M	V, M	128	72.0	59.2	92.0	88.8
SAT	V, P, M	P, M	128	72.0	61.0	92.0	89.3
SAT	V, P, M	V, P, M	192	**72.7**	**61.1**	93.0	89.4

Table 4
Accuracy comparison with the existing algorithms on ScanObjectNN dataset including realistic 3D shapes. (V: voxel, P: point set, M: multiview images, *: our implementation)

algorithm	3D shape representation		feature dim.	retrieval accuracy (MAP)	classification accuracy
	training	feature extraction			
Vconv-DAE*	V	V	6,192	33.7	72.8
FoldingNet*	P	P	512	33.6	65.9
MVAE*	M	M	30	21.7	—
SAT	V, P, M	V	64	37.3	69.0
SAT	V, P, M	P	64	37.1	70.2
SAT	V, P, M	M	64	36.9	71.3
SAT	V, P, M	V, P	128	38.0	72.3
SAT	V, P, M	V, M	128	39.0	73.0
SAT	V, P, M	P, M	128	38.9	73.1
SAT	V, P, M	V, P, M	192	**39.1**	**74.9**

[28], and VIP-GAN [46] obtain 3D shape features by using adversarial learning framework.

4.2. Experimental result

4.2.1. Comparison with existing 3D shape features

Table 3 and Table 4 compare accuracies of 3D shape features obtained by the proposed SAT and its competitors. Recall that the encoder parts of SAT act as a feature extractor for arbitrary combinations of 3D shape representations that SAT learned. Table 3 and Table 4 thus contain results of seven features produced by SAT, that are, features extracted from a single shape representation and "combined" features obtained by concatenating features due to two or three shape representations.

Table 3 as well as Table 4 indicate that SAT significantly outperforms the competitors especially in terms of retrieval accuracy. In Table 4, which shows results for ScanObjectNN dataset, SAT yields the best classification accuracy among the algorithms listed. In addition, the proposed feature is compact, i.e., it has only 192 dimensions even after concatenation. These results demonstrate that transcoding multiple shape representations is an effective way to learn expressive and compact features both of synthetic and realistic 3D shapes.

Superiority of SAT feature to its competitors is significant when they are evaluated under 3D model retrieval scenario. We suppose this is because training of SAT involves metric learning via optimization using the feature agglomeration loss L_{fa}. We will experimentally evaluate impact of the feature agglomeration loss on the retrieval accuracy in the next section.

4.2.2. In-depth evaluation of SAT

In this section, we empirically verify design parameters for SAT algorithm. Retrieval accuracy measured in MAP is used for evaluation. We set n_e at 64 unless otherwise stated.

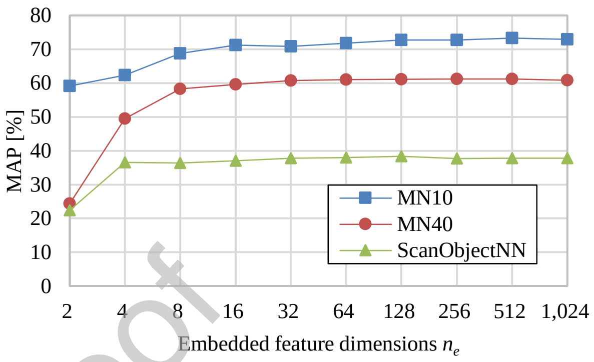


Fig. 2. Retrieval accuracy plotted against the number of dimensions n_e for the latent feature space.

Dimensions of learned shape feature: Fig. 2 plots retrieval accuracies against dimensions n_e for the latent feature space formed by SAT. Since we used the concatenated feature of three input shape representations, actual number of feature dimensions is 3×n_e. For all the datasets we used in the experiment, learned 3D shape features perform well in matching 3D shapes even with low dimensional feature embedding (e.g., n_e=8).

Architecture of encoder DNNs: This subsection evaluates influence that the network architecture of the shape encoders has on retrieval accuracy. To this end, we constructed four variants of SAT, whose encoder DNNs are wider, narrower, deeper, and shallower compared to the baseline architecture shown in Table 1. The wider or narrower encoder was constructed by doubling or halving the number of neurons in all the hidden layers of the baseline encoder DNNs. The deeper and shallower encoders were created as follows. For the voxel and multiview image encoders, we increased or decreased the number of residual blocks placed between the two max-pooling layers by one. For the point set encoder, we increased or decreased the number of fully-connected layers having 64 neurons by two layers. We fixed n_e at 64 for all the variants of SAT. Table 5 compares retrieval accuracies of SATs having different encoder architectures. All the SATs yields high MAP score of more than 60%. This result suggests that our SAT works robustly against changes of the encoder architecture, especially changes in the number of neurons and the number of layers of encoder DNNs.

Table 5
Influence that encoder architectures of SAT has on retrieval accuracy (ModelNet40 dataset).

encoder architecture of SAT	retrieval accuracy (MAP)
baseline shown in Table 1	61.1
wider	61.3
narrower	60.3
deeper	61.3

shallower	60.9

Retrieval among different shape representations: SAT learns the latent feature space shared by multiple shape representations. Thus, the learned feature can be used for comparison between different shape representations. Table 6 shows retrieval accuracies when retrieval target 3D models are queried by 3D models having different shape representation from the targets. Accuracies vary only small amount, regardless of if the paired representations are identical or not.

Table 6
Retrieval accuracy (MAP) between same and different shape representation pairs (MN40 dataset).

query	target shape representation		
shape representation	voxel	point set	multiview images
voxel	58.7	58.2	58.6
point set	58.2	58.7	58.9
multiview images	58.7	59.2	60.5

Table 7
Ablation study on SAT (MN40 dataset).

transcoding	feature agglomeration	data augmentation	shape representation		retrieval accuracy (MAP)
			training	feature extraction	
Yes	Yes	Yes	V, P, M	V, P, M	**61.1**
Yes	Yes	No	V, P, M	V, P, M	52.8
Yes	No	Yes	V, P, M	V, P, M	59.4
No	No	Yes	V, P, M	V, P, M	55.8
Yes	Yes	Yes	V, P	V, P	59.0
Yes	Yes	Yes	V, M	V, M	53.3
Yes	Yes	Yes	P, M	P, M	59.3
No	No	Yes	V	V	51.2
No	No	Yes	P	P	54.1
No	No	Yes	M	M	49.5

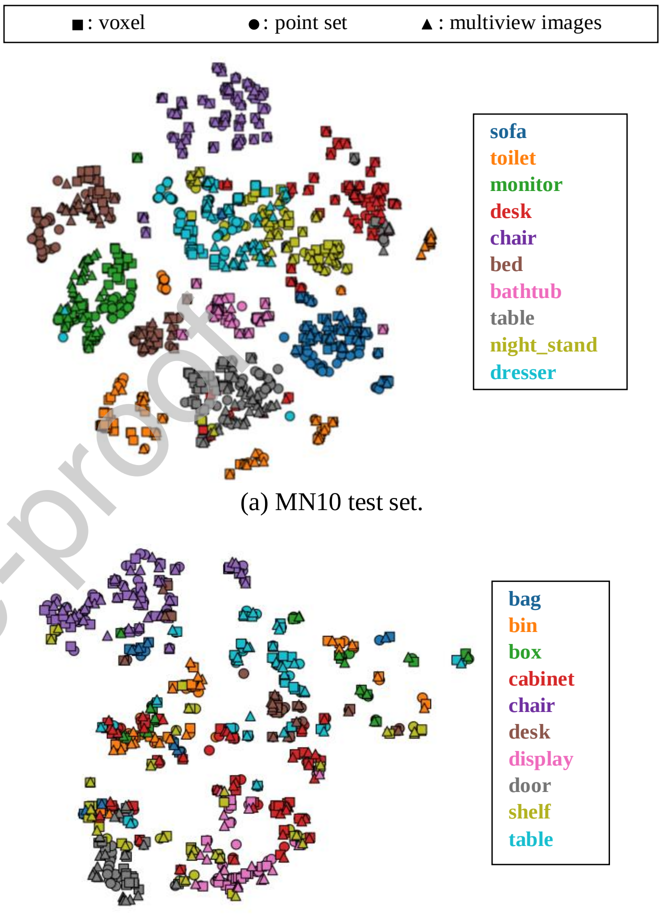
Visualization of embedded feature space: Fig. 3 visualizes the latent feature space learned by SAT. We used t-SNE algorithm [42] for visualization. In Fig. 3(a), embedded features of MN10 dataset are fairly well-separated into clusters by object category. In addition, features of objects in a class encoded from different shape representations are clustered close together. Such a latent feature space enables comparison among different shape representations as demonstrated in Table 6. Fig. 3(b) is the learned feature space formed by the realistic 3D shapes in ScanObjectNN. In Fig. 3(b), some feature clusters consist of multiple object categories. Apparently, transcoding realistic 3D shapes that have holes and cracks on their surface is more difficult than transcoding 3D CAD models.

Ablation study: Table 7 summarizes results of ablation study on SAT using the retrieval scenario. We evaluated efficacy of three components of SAT, that are, transcoding, feature agglomeration, and data augmentation. To disable transcoding, the permutation matrix **P** in the feature shuffling layer is fixed to an identity matrix during SAT training. To disable feature agglomeration, the term L_{fa} is omitted from the overall objective function. Table 7 indicates each of the three components is essential to learn accurate 3D shape feature.

We also evaluated the effectiveness of using three shape representations (i.e., voxel, point set, and multiview images) for training of SAT. As shown in Table 7, reducing the number of shape representations for training significantly decreases MAP of learned 3D shape feature. We speculate accuracy of 3D shape feature of SAT can be improved further if additional 3D shape representations (e.g., manifold mesh) and their corresponding encoders and decoders are available for training.

Shape transcoding: Fig. 4 exemplifies transcoded 3D shapes produced by the decoders of trained SAT. We fed a single shape representation of testing 3D model in MN40 into SAT. The embedded feature of the input is fed into all the decoders to reconstruct 3D shapes. Despite some geometrical error, each input shape is reconstructed into 3D shapes in three representations yet having the same object category as the input.

Computational efficiency: For all the experiments reported



(a) MN10 test set.

(b) ScanObjectNN test set.

Fig. 3. Visualization of embedded features of the 3D shapes in the test set of (a) MN10 and (b) ScanObjectNN. Color of plot indicates object category and shape of plot corresponds to input 3D shape representation. For clarity, (b) plots the features belonging to 10 out of 15 categories in ScanObjectNN.

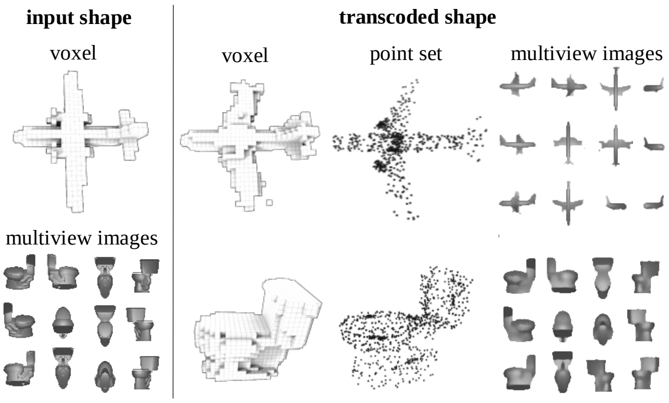


Fig. 4. Examples of shape transcoding by SAT (MN40 test set).

above, we used a PC having an Intel Core i7 6700 CPU, an Nvidia GeForce GTX 1080Ti GPU, and 64GB RAM. Our SAT fits in a common desktop PC since SAT consumed 6.7GB of RAM and 2.5GB of VRAM during training on MN40 dataset. Iterating the training of SAT for 200 epochs took about 11 hours. Feature extraction from one minibatch including eight shape tuples took about 0.04 seconds, which would be sufficiently fast for the purpose of shape retrieval and shape classification.

5. Conclusion and future work

In this paper, we introduced the idea of transcoding randomly across multiple 3D shape representations to obtain accurate and compact 3D shape feature under the constraint of unsupervised learning. We implemented the idea as a DNN called Shape Auto-Transcoder (SAT). SAT consists of a dedicated DNN architecture combined and a set of training objectives designed to learn expressive 3D geometric feature. SAT is trained so that it could transcode across multiple 3D shape representations such as voxel, 3D point set, and multiview 2D images of a 3D shape. In the process, the SAT synergistically captures expressive shape feature from multiple shape representations of the 3D shape. We verified efficacy of 3D shape feature learned by SAT through the experiments of 3D shape retrieval and 3D shape classification. SAT produces 3D shape feature more accurate and more compact than the existing 3D shape features learned in an unsupervised manner.

As future work, we will increase the number of 3D shape representations for training of SAT. We consider using, for example, singly-connected manifold mesh or polygon soup model to incorporate richer information about 3D geometry into SAT training. We also intend to further explore the architecture of the encoder/decoder DNNs for SAT to obtain better learned 3D shape features.

References

[1] C. R. Qi, H. Su, K. Mo, L. J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, *Proc. CVPR 2017*, 2017, 652–660.

[2] D. Maturana, S. Scherer, Voxnet: A 3d convolutional neural network for real-time object recognition, *Proc. IROS 2015*, 2015, 922–928.

[3] J. Masci, D. Boscaini, M. M. Bronstein, P. Vandergheynst, Geodesic Convolutional Neural Networks on Riemannian Manifolds, *Proc. ICCV Workshop 2015*, 2015, 37–45.

[4] H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, Multi-view convolutional neural networks for 3D shape recognition, *Proc. CVPR 2015*, 2015, 945–953.

[5] A. Sharma, O. Grau, M. Fritz, VConv-DAE: Deep Volumetric Shape Learning Without Object Labels, *Proc. ECCV 2016 Workshops*, 2016, 236–250.

[6] Y. Yang, C. Feng, Y. Shen, D. Tian, FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation, *Proc. CVPR 2018*, 2018, 206–215.

[7] Z. Zhu, X. Wang, S. Bai, C. Yao, X. Bai, Deep Learning Representation using Autoencoder for 3D Shape Retrieval, *Neurocomputing*, Vol. 204, 2016, 41–50.

[8] B. Leng, S. Guo, X. Zhang, Z. Xiong, 3D object retrieval with stacked local convolutional autoencoder, *Signal Processing*, Volume 112, 2015, 119–128.

[9] G. E. Hinton and R. S. Zemel. Autoencoders, minimum description length and helmholtz free energy, *Proc. NIPS 1994*, 1994, 3–10.

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, *Proc. NIPS 2014*, 2014, 2672–2680.

[11] A. Radford, L. Metz, S. Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, *Proc. ICLR 2015*, 2015.

[12] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, *Proc. NIPS 2016*, 2016, 2234–2242.

[13] C. Doersch, A. Gupta, A. A. Efros, Unsupervised Visual Representation Learning by Context Prediction, *Proc. ICCV 2015*, 2015, 1422–1430.

[14] G. Larsson, M. Maire, G. Shakhnarovich, Learning representations for automatic colorization, *Proc. ECCV 2016*, 2016, 577–593.

[15] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, Discriminative unsupervised feature learning with convolutional neural networks, *Proc. NIPS 2014*, 2014, 766–774.

[16] Z. Wu, Y. Xiong, S. Yu, D. Lin, Unsupervised Feature Learning via Non-Parametric Instance-level Discrimination, *Proc. CVPR 2018*, 2018, 3733–3742.

[17] H. ElNaghy, S. Hamad, M. E. Khalifa, Taxonomy for 3D content-based object retrieval methods, *IJRRAS*, **14**(2), 2013, 412–446.

[18] R. M. Rustamov, Laplace-Beltrami eigenfunctions for deformation invariant shape representation, *Proc. SGP 2007*, 2007, 225–233.

[19] M. M. Bronstein, I. Kokkinos, Scale-invariant heat kernel signatures for non-rigid shape recognition, *Proc. CVPR 2010*, 2010, 1704–1711.

[20] E. Wahl ; U. Hillenbrand ; G. Hirzinger, Surflet-pair-relation histograms: a statistical 3D-shape representation for rapid classification, *Proc. 3DIM 2003*, 2003, 474–481.

[21] M. Körtgen, M. Novotni, R. Klein, 3D Shape Matching with 3D Shape Contexts, *Proc. CESCG 2003*, 2003, Vol. 3, 5–17.

[22] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors, *Proc. SGP 2003*, 2003, 156–164.

[23] M. Novotni, R. Klein, 3D zernike descriptors for content based shape retrieval, *Proc. ACM symposium on Solid modeling and applications (SM)*, 2003, 216–225.

[24] D. Y. Chen, X. P. Tian, Y. T. Shen, M. Ouhyoung, On Visual Similarity Based 3D Model Retrieval, *CGF*, **22**(3), 2003, 223–232.

[25] R. Ohbuchi, K. Osada, T. Furuya, T. Banno, Salient Local Visual Features for Shape-Based 3D Model Retrieval, *Proc. SMI 2008*, 2008, 93–102.

[26] A. Brock, T. Lim, J.M. Ritchie, N. Weston, Generative and Discriminative Voxel Modeling with Convolutional Neural Networks, *Proc. NIPS 2016*, 2016.

[27] J. Wu, C. Zhang, T. Xue, W. T. Freeman, J. B. Tenenbaum, Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling, *Proc. NIPS 2016*, 2016.

[28] P. Achlioptas, O. Diamanti, I. Mitliagkas, L. Guibas, Learning Representations and Generative Models for 3D Point Clouds, *Proc. ICLR Workshops 2018*, 2018.

[29] K. Wang, Q. Yin, W. Wang, S. Wu, L. Wang, A Comprehensive Survey on Cross-modal, Retrieval, *arXiv preprint*, 2016, arXiv:1607.06215.

[30] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A. Y. Ng, Multimodal deep learning, *Proc. ICML 2011*, 2011, 689–696.

[31] F. Feng, X. Wang, R. Li, Cross-modal Retrieval with Correspondence Autoencoder, *Proc. MM 2014*, 2014, 7–16.

[32] V. Vukotić, C. Raymond, G. Gravier, Bidirectional Joint Representation Learning with Symmetrical Deep Neural Networks for Multimodal and Crossmodal Applications, *Proc. ICMR 2016*, 2016, 343–346.

[33] R. Zhang, P. Isola, A. A. Efros, Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction, *Proc. CVPR 2017*, 2017, 1058–1067.

[34] R. Kuga, A. Kanezaki, M. Samejima, Y. Sugano, Y. Matsushita, Multi-task Learning using Multi-modal Encoder-Decoder Networks with Shared Skip Connections, *Proc. ICCV 2017*, 2017, 403–411.

[35] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, *Proc. CVPR 2016*, 2016, 770–778.

[36] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, *Proc. ICML 2015*, 2015, 448–456.

[37] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, *Proc. ICML 2010*, 2010, 807–814.

[38] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, *Proc. ICLR 2015*, 2015.

[39] K. He, X. Zhang, S. Ren, J. Sun, Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, *Proc. ICCV 2015*, 2015, 1026–1034.

[40] Z. Wu et al., 3D ShapeNets: A Deep Representation for Volumetric Shape Modeling, *Proc. CVPR 2015*, 2015, 1912–1920.

[41] A. X. Chang et al., ShapeNet: An Information-Rich 3D Model Repository, *arXiv preprint*, 2015, arXiv:1512.03012.

[42] L. Maaten, G. Hinton, Visualizing Data using t-SNE, *JMLR*, 2008, Vol. 9, 2579–2605.

[43] M. A. Uy et al., Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data, *Proc. ICCV 2019*, 2019, 1588–1597.

[44] P. Cignoni et al., MeshLab: an Open-Source Mesh Processing Tool, *Proc. Sixth Eurographics Italian Chapter Conference*, 2008, 129–136.

[45] Y. Wang et al., An efficient and effective convolutional auto-encoder extreme learning machine network for 3d feature learning, *Neurocomputing*, Vol. 174, Part B, 2016, 988–998.

[46] Z. Han et al. View inter-prediction gan: Unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions, *Proc. AAAI 2019*, 2019, 8376–8384.

[47] F. Bernardini et al., The ball-pivoting algorithm for surface reconstruction, *TVCG*, **5**(4), 1999, 349–359.

Author Declaration

We confirm that there are no known conflicts of interest associated with this manuscript and significant financial support for this work that could have influenced its outcome. We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us. We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.