Reconstruction or Recognition: Justifying Single-view 3D Reconstruction Networks

Anonymous ECCV submission

Paper ID 6890

Abstract. Due to the effectiveness in learning and approximating nonlinear functions, deep neural networks have received much attention from the 3D vision community and are expected to perform well on the task of reconstructing 3D nonlinear shapes (e.g., represented by voxels or point clouds). However, recent works have shown that these deep neural networks tend to perform *recognition* instead of 3D reconstruction, causing us to rethink the effectiveness of neural networks in such tasks. In this paper, we show that the bias towards recognition is due to the intrinsic properties of dataset: when the training set of 3D shapes has a more clustered structure, the deep neural networks trained on this dataset become more likely to perform recognition than reconstruction. We provide both qualitative and quantitative results to show this strong correlation between the dataset and deep neural networks, by visualizing the dataset in low dimensions, and by defining novel ways to measure the clustering coefficient of 3D datasets. Guided by these observations and experiment results, we show that the standard deep neural networks trained on ShapeNet tend to perform reconstruction instead of recognition.

1 Introduction

Using deep learning for single-view 3D reconstructions continue to be of interest. While numerous papers have presented innovative deep learning architectures to advance the state-of-the-art in 3D reconstruction tasks [14, 18, 5, 25, 7, 38, 31, 22, 28, 32, 35], there are very few works that attempt to address the fundamental properties of these tasks. Indeed, the problem of reconstructing 3D irregular shapes has become a new machine learning paradigm, and the rule of thumbs for practitioners to process data and train deep neural networks are often different from those of learning on regular data (e.g., Adam [11] is often used instead of SGD). Therefore, how do neural networks learn in this new paradigm of 3D reconstruction learning, in a different way from conventional vector classification and regression problems?

Recently, the authors in [26] presented a surprising viewpoint to the question above. They experimentally showed that cutting-edge deep neural networks for 3D reconstruction tasks tend to focus more on *recognition* than on *reconstruction*. That is, these neural networks tend to predict a 3D shape by first classifying the input image into a certain cluster (which not necessarily corresponds to the label of the 3D shape), and then generating the mean shape of the corresponding cluster. The major experimental evidence of this claim is that the quality of

the 3D predictions of these deep neural networks cannot be distinguished from that of purely clustering-based and retrieval-based baselines. This is an interesting observation, because it indicates that for the task of 3D reconstructions, deep neural networks tend to memorize the mean shape and correlate that to the semantic meaning of the image input, instead of generating a shape using geometric ways, e.g., by merging fine-grained local parts into a global shape. If the claim is true, it means that for the task of 3D reconstruction, cutting-edge deep neural networks tend to perform more of a memorization task than a generalization task [2].

In this work, we show both quantitatively and qualitatively that the conclusion of [26] is not universal, and is a complicated consequence of inappropriate data set collection and improper data usage. We propose to measure the "clustering-tendency" of a dataset using well-established concepts in data mining, namely affinity propagation [30] and silhouette score [29], and we show that when the datasets used for training 3D reconstruction tasks do not form into clustered patterns, the resulting neural networks can learn to focus more on reconstruction than recognition. More importantly, we show that even for real datasets like ShapeNet [3], the neural networks trained on it still perform a reconstruction based task. Our interpretation of recognition vs reconstruction is intellectually diverse, in that it relates 3D reconstruction to both the data (silhouette score) and the training procedures. It is also practically useful, in that it provides guidance to 3D dataset collections. More specifically, we show that the 3D shapes in the training set should contain more diversity than the training images, in order to avoid generating a recognition-based machine learning model.

1.1 Related works

Whether deep neural networks perform memorization or generalization has been of primary concern in modern machine learning. The well-known conjecture, similar to what we will present, is that the optimization process is "content-aware" and depends on the properties of data itself [2]. It is also shown in [2] that certain regularization techniques during training help the deep neural networks to generalize instead of memorizing a task. For 3D shape reconstructions, [26] show that neural networks tend to memorize the mean shape instead of reconstructing in a geometric sense. Indeed, many works have also shown the effectiveness of mean shapes and recognition information in improving the quality of 3D reconstructions [10, 20, 13].

In contrast, there are also many works that utilize the distributional information in the continuous latent space of 3D shapes [15, 8, 21, 14, 34, 36, 37] to improve 3D reconstructions, which are beyond the recognition-based framework. Notably, [33, 6] showed that shape arithmetic can be done in the latent space of 3D shapes, ruling out the possibility that neural networks perform only recognition in their problem settings (because performing arithmetic requires more than the information of the mean shapes of discrete clusters). Some other works propose to generate 3D shapes by decomposing each shape into parts [27, 16, 24] or into a continuous process [23, 4], which are also beyond a simple recog-

nition task. However, the authors of [33, 1, 38] treat 3D reconstruction based auto-encoding as the backbone of unsupervised classification on point clouds. Although the input data in these works are 3D shapes, instead of 2D images, the fact that shape information helps classification does seem to intensify the belief that 3D reconstruction focuses more on recognition than reconstructions. Nonetheless, auto-encoding is not only restricted to 3D reconstructions, and it has become the default backbone in many other representation learning tasks as well. Thus, the fact that shape information helps recognition cannot be a primary reason to conclude what 3D-reconstruction networks learn. A meaningful future direction is to investigate the difference between the learned functions of 3D reconstructions used respectively for unsupervised classification and supervised shape generation, as the main goals of these two directions are not exactly the same.

Note that the 3D reconstruction problem can be viewed as a specific case of the more general *distribution learning* [17, 19] problem. However, unlike theoretical works on distribution learning that extend kernel methods to regressing distributions, our work focuses on deep learning. Nonetheless, it is interesting to see that classical works on distribution learning [17, 19] treat the output distribution as a continuous linear combination on all samples of training distributions directly, instead of using a two-step way of predicting the cluster index followed by predicting the "mean distribution".

2 Problem formulation

To start, we formally define the problem on *reconstruction* vs *recognition* (see Section 2.2). We will show that whether the trained neural network biases itself towards either of these two different learning paradigms depends on the property of datasets used for training. Thus, we also need to define metrics to measure the training data (see Section 2.3).

2.1 Preliminaries

The problem that we study concerns the reconstruction of a 3D shape from a single image. The input I is a 2D image. The output S is represented by a point cloud. We do not consider voxel-based volumetric representations in this work. For the point-cloud-based representations, each shape S is a point set that contains a set of 3D points. A neural network model f is trained to predict the shape S from the input image I, by minimizing the empirical loss defined for certain loss function l,

$$\min_{f} \sum_{i=0}^{n-1} l(f(I_i), S_i). \tag{1}$$

We would like to study the property of the function f to see if it performs more of a recognition task or a reconstruction task.

Two metrics are used to evaluate the difference between the reconstruction $\widehat{S} = f(I)$ and the ground-truth point cloud S, namely Chamfer distance and F-score.

Chamfer distance. Chamfer distance measures the total distance between one point set and the other one by searching for the nearest points in the other set.

$$d_{CH}(S, \widehat{S}) = \frac{1}{|S|} \sum_{\mathbf{x} \in S} \min_{\widehat{\mathbf{x}} \in \widehat{S}} \|\mathbf{x} - \widehat{\mathbf{x}}\|_2 + \frac{1}{|\widehat{S}|} \sum_{\widehat{\mathbf{x}} \in \widehat{S}} \min_{\mathbf{x} \in S} \|\widehat{\mathbf{x}} - \mathbf{x}\|_2. \tag{2}$$

Although Chamfer distance is effective and simple to compute, it may be strongly influenced by a small number of outlier points in the point cloud representation. Thus, we also adopt the F-score on predicting point sets as suggested by [26].

F-score. An alternative way to evaluate shape reconstruction is F-score, which is the harmonic mean of precision and recall. The precision *Prec* of the reconstruction $\widehat{S}$ given the corresponding ground truth S, for a fixed distance threshold d, is defined as:

$$Prec(d, S, \widehat{S}) = \frac{1}{|\widehat{S}|} \sum_{r \in \widehat{S}} [\![\min_{s \in S} ||r - s|| < d]\!], \tag{3}$$

where $[\![\cdot]\!]$ is the Iverson bracket.

Similarly, the recall *Rec* of the ground truth S from the reconstruction $\widehat{S}$ is:

$$Rec(d, S, \widehat{S}) = \frac{1}{|S|} \sum_{s \in S} [\![\min_{r \in \widehat{S}} ||s - r|| < d]\!]. \tag{4}$$

Using these two quantities, F-score is calculated using the following:

$$F(d, S, \widehat{S}) = \frac{2 \times Prec(d, S, \widehat{S}) \times Rec(d, S, \widehat{S})}{Prec(d, S, \widehat{S}) + Rec(d, S, \widehat{S})}. \tag{5}$$

The accuracy of reconstruction is quantified by precision, which measures how closely the reconstructed points lie to the ground truth. The completeness is quantified by recall, which on the other hand measures how much the ground truth shape is being covered by the reconstructed shape. Therefore, a high F-score indicates that the reconstruction is both accurate and complete [12].

2.2 Recognition or reconstruction

In this section, we formally define the two learning paradigms studied in the paper, namely *recognition* and *reconstruction*.

Definition 1. *(Recognition neural network) A recognition-based neural network f predicts the shape reconstruction in two steps. That is, the neural network reconstruction function can be written as*

$$\widehat{S} = f(I) = f_1(f_2(I)), \tag{6}$$

in which the function $f_2(\cdot)$ maps the input image I to a scalar index $f_2(I)$, and $f_1(f_2(I))$ maps this scalar index to the mean shape of the particular cluster with index $f_2(I)$.

Definition 2. *(Reconstruction neural network) A reconstruction-based neural network directly generates the 3D reconstruction. That is,*

$$\widehat{S} = f(I), \tag{7}$$

and the reconstruction does not explicitly use any information of the clusters of images.

Since the most popular ways in single-image 3D reconstruction often have an encoder-decoder structure, a similar (informal) definition is that the codewords obtained from the encoder do not form into clusters.

Main problem In this paper, we study whether neural networks perform recognition or reconstruction. We show that the tendency towards either of these two depends on the property of data.

Note that the main statement of [26] is that the deep neural networks for single-image-reconstructions of 3D shapes mainly perform recognition tasks. In other words, the function of the neural network is closer to Definition 1 than to Definition 2.

2.3 Clustering tendency of datasets

In this section, we define the metric that we use to measure the clustering tendency of a dataset. In particular, we use silhouette score [29] to measure the clustering tendency. Given a dataset $D = \{x_i\}_{i=0}^{N-1}$ and an arbitrary distance $d(x, y)$ function[1], we can determine a *clustering* of the dataset by specifying a clustering function $C(\cdot)$. For each sample x_i, the clustering function gives the cluster label $C(x_i)$. We use C_i to denote the cluster that contains x_i, i.e., the cluster label of C_i equals $C(x_i)$. Sometimes, the dataset already contains the ground truth clustering. More often, the clustering has to be obtained by an algorithm. Then, the silhouette score of the i-th sample is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}, \tag{8}$$

where

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, j \neq i} d(i, j), \qquad b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j). \tag{9}$$

The clustering tendency, defined using the silhouette score, is given by:

$$\textit{Clustering-tendency} = \frac{1}{N} \sum_{i} s(i). \tag{10}$$

[1] We consider any distance function that satisfies the following three properties $d(x_1, x_2) = d(x_2, x_1)$, $d(x_1, x_2) \geq 0$, and $d(x, x) = 0$. In our experiment, we use Chamfer distance as the distance metric for point sets. We use ℓ-1 distance to measure the distance between images.

For dataset without ground truth clusters, we have to find the clustering function $C(\cdot)$. In this paper, we use affinity propagation [30] for clustering. Affinity propagation is suitable for our setting, because it does not require the number of clusters to be determined beforehand, and it uses the predetermined distance metric to assign each data point x_i to a cluster C_i. We use the Chamfer distance in (2) as the distance function $d(x, y)$ in this clustering to handle unordered point clouds in our datasets.

3 Analysis of an interpolation-based synthetic dataset

In this section, we first theoretically analyze the connection between the clustering tendency of training data, and the bias of the trained neural networks towards either recognition or reconstruction (see the definition in Section 2.2). Then, we use a synthetic dataset to support the theoretical analysis (see Section 3.2 and Section 3.3).

3.1 Theoretical analysis

In this subsection, we analyze the underlying intuition behind the recognition phenomenon observed in 3D reconstruction tasks. We show that, when the clustering tendency of the training point clouds is larger than that of the training images, the learned model tends to perform recognition than reconstruction. Consider Figure 1.

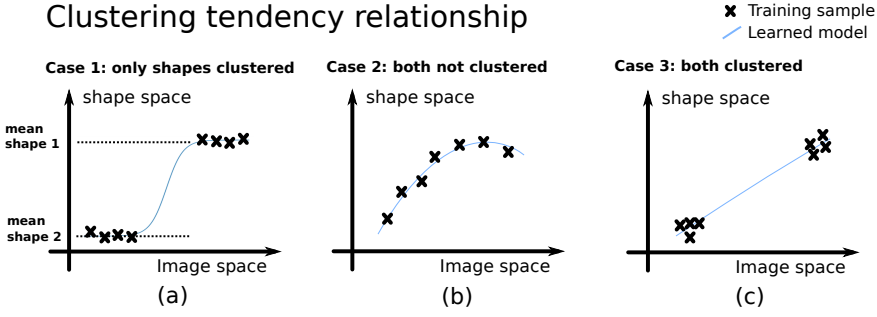


Fig. 1: When the training images are less clustered than the training shapes (as shown in subfigure (a)), the learned model shows the tendency of recognition.

In Figure 1, we illustrate three different types of training data. In the subplot (a), the training images are less clustered than the training shapes. In this case, the model trained on this type of data exhibits a higher tendency towards recognition, i.e., the shape predictions concentrate on the mean shapes, illustrated as the intersections between the dashed lines and the y-axis. In the subplot (b), neither the training images nor the training shapes show a strong tendency towards clustering. Thus, the model learned on this type of data does not show the tendency towards clustering, either. In the subplot (c), we show another case in which even if both images and shapes are clustered, the learned model does not show a high clustering tendency. This may be counter-intuitive, but it indeed matches what we observe in our experiments.

To summarize, we list three data types that we explore in our experiments on image-to-shape reconstructions.

- Case 1: Training shapes are clustered, while training images are not.
- Case 2: Neither training images nor training shapes are clustered.
- Case 3: Both training images and training shapes are clustered.

Our informal claim to the question of whether neural networks are biased towards *recognition* or *reconstruction* is the following:

(Main claim:) *Deep-learning-based single-view 3D reconstructions tend to use recognition only in Case 1, and use reconstruction in Case 2 and Case 3 instead.*

3.2 Synthetic data generation

In this subsection, we provide the details of the synthetic shape datasets. We use the software Blender to generate base shapes in a mesh format. Then, we use the Shrinkwrap modifier in "Nearest Vertex" mode to define the shape morphing between the two base shapes and control the interpolation progress by the Blender Shape Keys panel. After creating the base dataset, we render training images and sample point clouds from mesh to form training sets that cover different cases considered in Section 3.1, namely Case 1, 2, and 3.

Base dataset # 1 (Cube-sphere) The first base dataset is obtained by interpolating between a cube and a sphere of a similar size. See Figure 2. We interpolate 1000 intermediate shapes between these two ends. For each intermediate shape, we obtain both the image from the isometric view and a point-cloud representation of 1024 3D points. The 1000 intermediate shapes are divided into 700 training shapes, 200 testing shapes, and 100 validation shapes randomly.

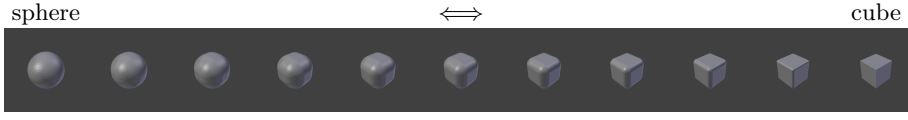
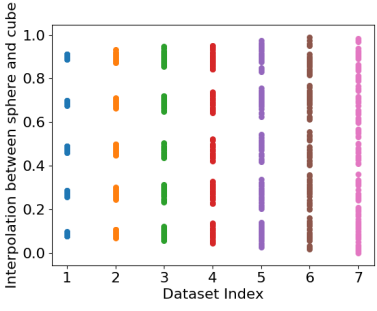


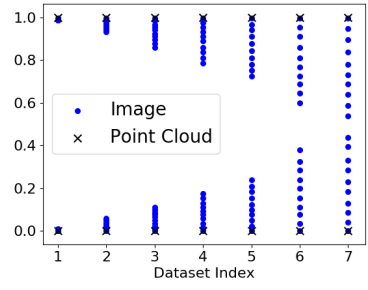
Fig. 2: Base dataset # 1: interpolation between a sphere and a cube.

Subsampled dataset 1.1 for case 2 and case 3 Subsampled dataset 1.1 aims to cover the transition between Case 2 and Case 3. This dataset is a collection of 7 datasets 1.1.n (n = 1, 2, ...7). Each dataset 1.1.n contains five clusters of shapes (and the corresponding images). From the subsampled dataset 1.1.1 to the subsampled dataset 1.1.7, the clustering tendency becomes increasingly lower. More specifically, the subsampled dataset 1.1.1 has five clearly separated clusters (see the first vertical line in Figure 3a), while the subsampled dataset 1.1.7 almost continuously covers the entire transition from a sphere to a cube (see the last vertical line in Figure 3a). It should be noted that all the seven subsampled datasets contain the same number of training images, i.e., each one contains five clusters, and each cluster contains 20 samples. However, the test set of each subsampled dataset remains the same as the test set in the base dataset 1. In other words, we only subsample the training set, while maintaining a fixed test set.

Subsampled dataset 1.2 for case 1 and case 3 Subsampled dataset 1.2 is another subsampled version of the base dataset 1. This dataset aims to cover the transition from Case 1 to Case 3. This dataset is a collection of 7 subsampled datasets 1.2.n (n = 1, 2, ...7). Each dataset 1.2.n contains two clusters of samples. However, different from the subsampled dataset 1.1, although each (image, shape) training pair in the subsampled dataset 1.2 has a distinct 2D image, the corresponding shape can only be one of the two end shapes (i.e., the two base shapes cube and sphere), and no intermediate shape exists. Thus, in this case, the clustering tendency of the training shapes is always high, while the clustering tendency of the training images becomes lower and lower from the subsampled dataset 1.2.1 to the subsampled dataset 1.2.7. Again, the training images of the subsampled dataset 1.2.7 almost continuously covers the entire transition from the image of a sphere to a cube. Thus, the subsampled dataset 1.2.1 represents an example of Case 3, while the subsampled dataset 1.2.7 represents an example of Case 1, and the collections of datasets 1.2.n (n = 1, 2, ...7) represent the transition from Case 3 to Case 1. In each dataset, the number of training images is 2×10, where 2 is the number of clusters, and 10 is the number of samples in each cluster. Similar to the subsampled dataset 1.1, the test set of the subsampled dataset 1.2 is still the fixed one in the base dataset 1.



(a) Dataset 1.1.n (n = 1, 2, ...7). Different subsampled datasets are generated to have different clustering tendencies.



(b) Dataset 1.2.n (n = 1, 2, ...7). The dot markers represent image distribution, while the cross markers represent point cloud distribution.

Fig. 3: Coverage of two subsampled data sets 1.1.n and 1.2.n.

Base dataset #2 (Cube-pyramid-sphere) To complement the first dataset, we generate another dataset that interpolates between three base shapes, namely cube, pyramid, and sphere. These three shapes have a similar size. Between these three base shapes, we interpolate 10000 intermediate shapes. Similar to Base dataset #1, we create subsampled datasets from Base dataset #2 to cover different cases discussed in Section 3.1 as well. Due to the space limitation, we put the details of the base dataset generation, subset sampling, and the experiment results, in the supplementary materials.

3.3 Experiment results

In this subsection, we present the experimental results to support the main claim in Section 3.1 that recognition only happens if the training images are less clustered than the training shapes, i.e., Case 1. To validate the claim, we have constructed different datasets that cover the transitions between Case 1, 2, and 3 in Section 3.2. For the neural network model, we adopt an auto-encoder architecture, using ResNet-18 for images[9] as the encoder, a ResNet-based FoldingNet [38] as the decoder, and Chamfer distance as the loss function. For subsampled dataset 1.1, we train a separate model for each of the dataset from 1.1.1 to 1.1.7 on respectively for 1500 epochs, in order to ensure their learning capacity has saturated. For subsampled dataset 1.2, we train over each subsampled dataset 1.2.n for 3000 epochs, which is higher because the size of each subsampled dataset 1.2.n is small (20). During training, we use the Adam optimizer, an initial learning rate of 0.003, and learning rate decay of 0.1. We use the validation set to choose the best model for prediction, instead of using the test set.

Measuring the dataset metrics In this part, we show the quantitative results to support our main claim in Section 3.1. For each subsampled dataset, we plot the clustering tendency of input data versus that of output data. Since we have two types of input data, namely training images and training shapes, we plot the results for input-image-vs-prediction and input-shape-vs-prediction separately. The results are shown in Figure 4. The results on the first row concern subsampled dataset 1.1, i.e., the transition from case 3 to case 2, while the results on the second row concern subsampled dataset 1.2, i.e., the transition from case 3 to case 1. For the results in subsampled dataset 1.1, even if the clustering tendency of the input varies, the output clustering tendency remains the same. This result supports our theoretical analysis in Figure 1 (b) and (c), which is that the clustering tendency in both Case 2 and Case 3 is low. However, on the second row of Figure 2.3, we can see that the output clustering tendency clearly increases when the input image clustering tendency decreases. This is exactly the same as our theoretical analysis in Figure 1 (a), i.e., the clustering of output shapes only happens if the training images are not clustered while the training shapes are clustered. Note that on the second row, the training shapes only contain two end shapes (sphere and cube). Thus, the clustering tendency of training shapes is always high (see the right Figure on the second row).

Visualizing the distance matrix We also visualize the distance matrices of the different cases to support our main claim in Section 3.1. See Figure 5. In the left subfigure, we visualize the results for subsampled dataset 1.1 that covers case 3 and case 2. In the right subfigure, we visualize the results for subsampled dataset 1.2 that covers case 3 and case 1. The seven rows in the left subfigure correspond to subsampled datasets from 1.1.1 to 1.1.7. The seven rows on the right subfigure correspond to the subsampled datasets from 1.2.1 to 1.2.7. In each subfigure, the left column represents the distance matrices of the training images, the middle column represents those of the training point clouds, and the right volumn represents those of the predicted point clouds during testing. For

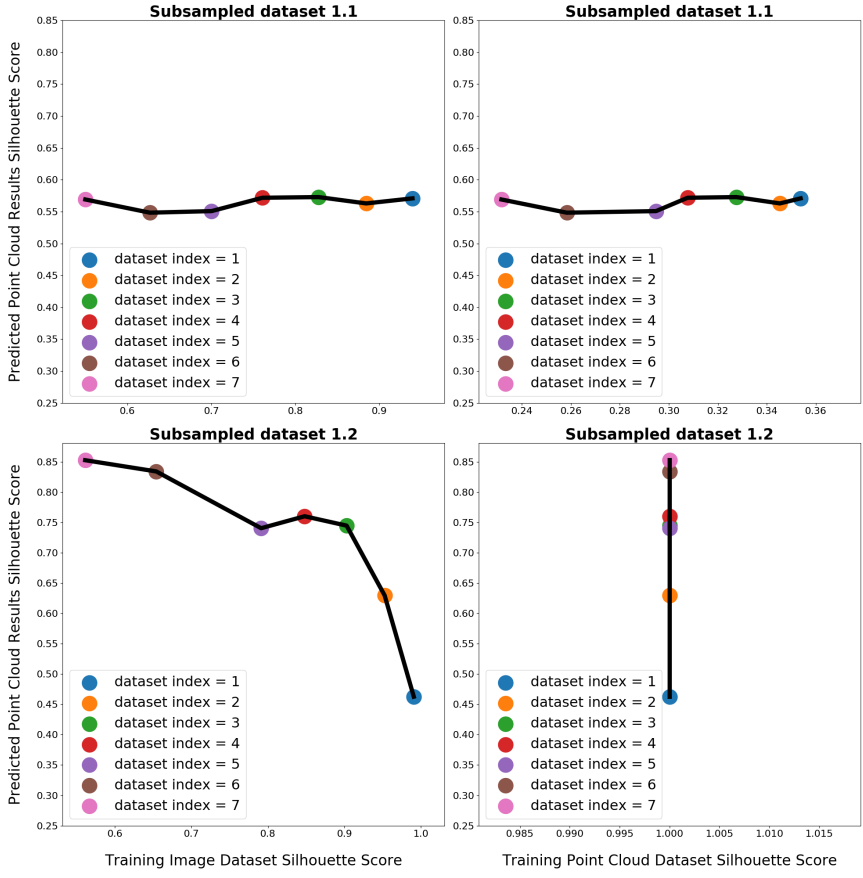


Fig. 4: Training-vs-prediction silhouette scores. Only Case 1 (the points with large dataset indices shown in the left-bottom figure) shows high tendency towards recognition.

training images, the distance is defined using image $\ell 1$-loss. For point clouds, the distance is the Chamfer distance.

From the distance matrices on the right column of each figure, we can see that the predicted point clouds form into clusters, only on the lower rows in the right figure. That is, the learned neural networks only perform recognition when the training dataset is close to Case 1. In other two cases (Case 2 and 3), the learned neural network tend to interpolate instead of concentrating on the mean shapes, which leads to a continuous change in the similarity matrix (e.g., see the third column).

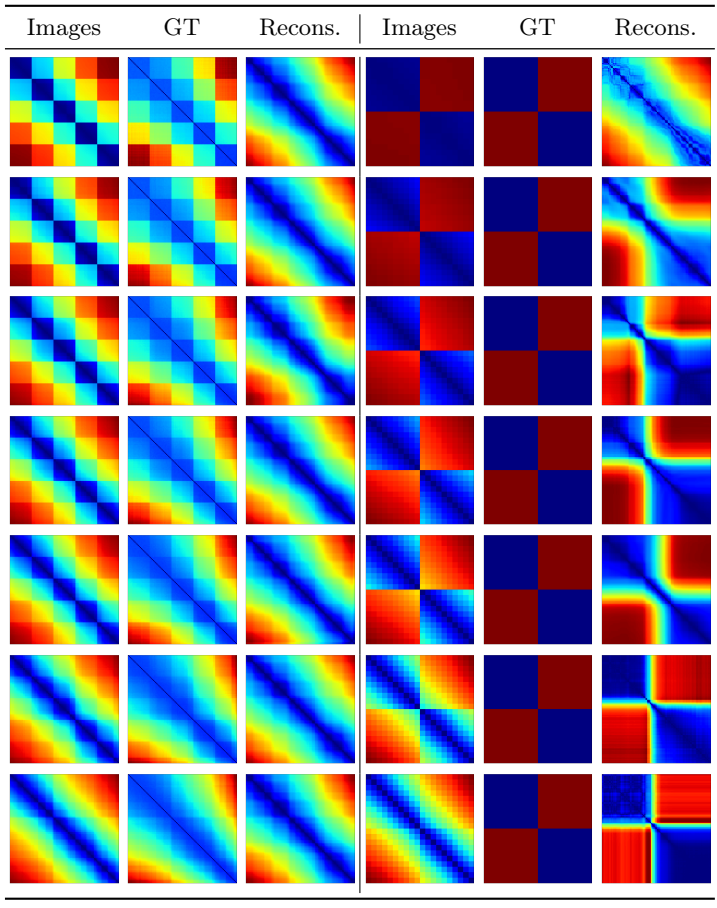


Fig. 5: Distance matrices. Blue represents small distance and red represents large distance. **(Left)** predicted shapes of subsampled dataset 1.1 are not clustered. **(Right)** predicted shapes of subsampled dataset 1.2 are clustered if the training images are not clustered (i.e., if the dataset belongs to Case 1). Top to bottom: dataset index from 1 to 7.

4 Analysis of recognition vs reconstruction in real dataset

In this section, we extend our analysis onto real datasets. We use the standard ShapeNet dataset [3] for our evaluation. We show that on real dataset like ShapeNet, even a standard 3D reconstruction deep neural network tends to perform a reconstruction task rather than a recognition task. Our conclusion is based on two observations.

– First, the Chamfer distance and F-score achieved by the standard 3D reconstruction network is close (within 1%) to that achieved by Oracle-Nearest-Neighbor (Oracle-NN) based reconstruction (see the definition in).

– Second, the 2D T-SNE visualization of the high-dimensional codewords, obtained during the auto-encoding training, does not exhibit a clear tendency to form clusters, compared to models trained with label information. However, the reconstruction performance is better without label information than that with label information (Figure 7).

4.1 Comparison with Oracle-NN

The Oracle-NN classifier [26] predicts a test point cloud $\widehat{S}$ from an input image I by searching for the training shape that has the smallest Chamfer distance to the ground-truth point cloud S. Since the Oracle-NN classifier requires the ground-truth point cloud S when searching for the nearest neighbor point cloud, it is impossible to be realized in practice. It is a theoretical baseline that characterizes the performance limit of any practical retrieval-based classifier. In [26], it was shown that Oracle-NN achieves a superior performance than deep-learning based reconstructions (which is expected). However, here, we show that a standard deep auto-encoding network, using ResNet for both encoding and decoding, can achieve almost exactly the same as Oracle-NN, both in terms of the Chamfer distance and the F-score. Consider Figure 6. We consider the auto-encoder trained in two settings. In the first setting ("weak label info"), the class labels are provided during the first 5 epochs of training to provide a good initialization. However, this label information is removed starting from the 6-th epoch, and not available any more in the remaining 95 epochs. In the second setting, we provide no class label information at all. In Figure 6, we can see that in both settings, especially the setting with weak label information, the Chamfer distance and F-score are almost exactly the same to those of Oracle-NN.

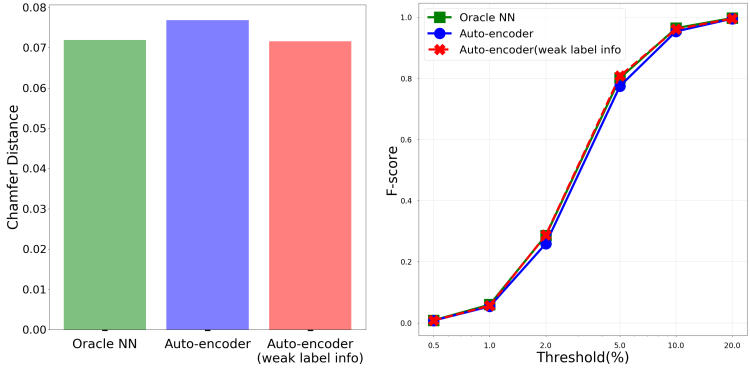
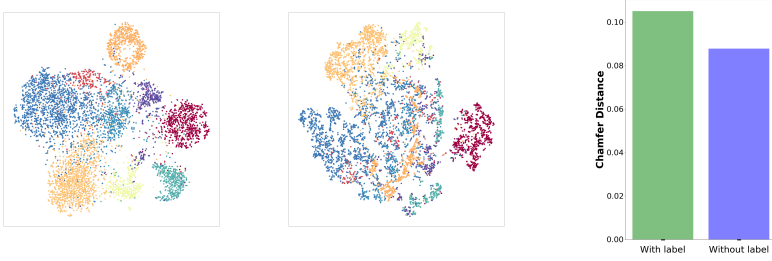


Fig. 6: This figure compares the performance of Oracle-NN to that of a standard ResNet auto-encoder based on FoldingNet. These two classifiers achieve almost exactly the same performance, despite the fact that Oracle-NN requires impractical oracle information unavailable during test time.

4.2 Visualization using T-SNE

In this part, we show that the standard neural network trained on ShapeNet performs more of a reconstruction task rather than recognition, by showing the 2D T-SNE plot (see Figure 7). The two plots in Figure 7 show the visualizations of a network trained with or without class label supervision (each model is trained for 100 epochs). Each 2D point in the T-SNE plot represents a codeword obtained during testing time. We only select the 9 classes with the most samples, so that the clustering effect can become more prominent if it indeed exists. The two neural networks have the same structure (FoldingNet with residual links), and they are trained using the same procedures (100 epochs, learning rate 0.003, batch size 32, weight decay 1E-6, and both with learning rate decay).

In the left figure, in which the neural network is trained with label supervision, we see a clear clustering pattern with higher Chamfer loss. In contrast, the right figure does not show a clear clustering pattern but with lower Chamfer loss. This comparison shows that classification information is not always useful for 3D reconstructions. The T-SNE comparison is also another piece of evidence to show that standard training does not lead neural networks to perform a recognition task.



(a) With labels (b) Without labels (c) Chamfer loss

Fig. 7: T-SNE and Chamfer loss of the same network trained with or without label information. The network trained without label information does not show a clear tendency to form clusters while yielding smaller Chamfer loss.

4.3 Qualitative results

In Figure 8, we provide the reconstruction results of the three schemes considered in Section 4.1.

5 Conclusions

In this paper, we study the question of whether single-view 3D reconstruction neural networks tend to perform recognition (memorization) or reconstruction

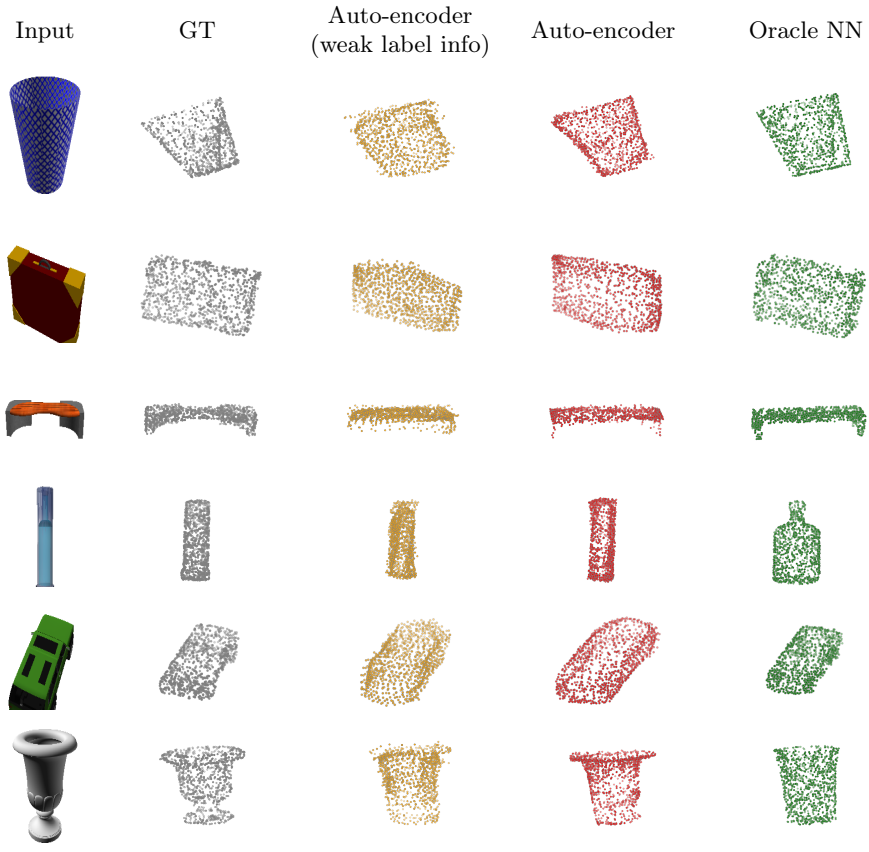


Fig. 8: Qualitative results show that standard auto-encoders achieve comparable reconstruction performance to Oracle-NN.

(interpolation). Our conclusion is that the tendency depends on the properties of the dataset that can be measured by the clustering tendency of training samples. In real datasets, we do observe that the neural networks tend to perform interpolation instead of memorization. In short, the recognition phenomenon arises from complicated properties of data, instead of purely from the deep learning models. A meaningful future work is to study deep learning techniques, from the perspectives of dataset collection, network architecture, and optimization procedures, to make neural network perform reconstruction even if the dataset is biased towards recognition.

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. arXiv preprint arXiv:1707.02392 (2017)
2. Arpit, D., Jastrzębski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M.S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al.: A closer look at memorization in deep networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 233–242. JMLR. org (2017)
3. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
4. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: European conference on computer vision. pp. 628–644. Springer (2016)
5. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 605–613 (2017)
6. Girdhar, R., Fouhey, D.F., Rodriguez, M., Gupta, A.: Learning a predictable and generative vector representation for objects. In: European Conference on Computer Vision. pp. 484–499. Springer (2016)
7. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A papier-mâché approach to learning 3d surface generation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 216–224 (2018)
8. Gwak, J., Choy, C.B., Chandraker, M., Garg, A., Savarese, S.: Weakly supervised 3d reconstruction with adversarial constraint. In: 2017 International Conference on 3D Vision (3DV). pp. 263–272. IEEE (2017)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (June 2016). https://doi.org/10.1109/CVPR.2016.90
10. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 371–386 (2018)
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
12. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (ToG) **36**(4), 1–13 (2017)
13. Kurenkov, A., Ji, J., Garg, A., Mehta, V., Gwak, J., Choy, C., Savarese, S.: Deformnet: Free-form deformation network for 3d shape reconstruction from a single image. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 858–866. IEEE (2018)
14. Li, C.L., Zaheer, M., Zhang, Y., Poczos, B., Salakhutdinov, R.: Point cloud gan. arXiv preprint arXiv:1810.05795 (2018)
15. Li, K., Garg, R., Cai, M., Reid, I.: Single-view object shape reconstruction using deep shape prior and silhouette. arXiv preprint arXiv:1811.11921 (2018)
16. Niu, C., Li, J., Xu, K.: Im2struct: Recovering 3d shape structure from a single rgb image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4521–4529 (2018)
17. Oliva, J., Póczos, B., Schneider, J.: Distribution to distribution regression. In: International Conference on Machine Learning. pp. 1049–1057 (2013)

18. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 165–174 (2019)
19. Póczos, B., Rinaldo, A., Singh, A., Wasserman, L.: Distribution-free distribution regression (2013)
20. Pontes, J.K., Kong, C., Sridharan, S., Lucey, S., Eriksson, A., Fookes, C.: Image2mesh: A learning framework for single image 3d reconstruction. In: Asian Conference on Computer Vision. pp. 365–381. Springer (2018)
21. Sinha, A., Unmesh, A., Huang, Q., Ramani, K.: Surfnet: Generating 3d shape surfaces using deep residual networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6040–6049 (2017)
22. Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J.B., Freeman, W.T.: Pix3d: Dataset and methods for single-image 3d shape modeling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2974–2983 (2018)
23. Sun, Y., Wang, Y., Liu, Z., Siegel, J.E., Sarma, S.E.: Pointgrow: Autoregressively learned point cloud generation with self-attention. arXiv preprint arXiv:1810.05591 (2018)
24. Sung, M., Su, H., Kim, V.G., Chaudhuri, S., Guibas, L.: Complementme: weakly-supervised component suggestions for 3d modeling. ACM Transactions on Graphics (TOG) **36**(6), 1–12 (2017)
25. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2088–2096 (2017)
26. Tatarchenko, M., Richter, S.R., Ranftl, R., Li, Z., Koltun, V., Brox, T.: What do single-view 3d reconstruction networks learn? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3405–3414 (2019)
27. Tulsiani, S., Su, H., Guibas, L.J., Efros, A.A., Malik, J.: Learning shape abstractions by assembling volumetric primitives. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2635–2643 (2017)
28. Tulsiani, S., Zhou, T., Efros, A.A., Malik, J.: Multi-view supervision for single-view reconstruction via differentiable ray consistency. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2626–2634 (2017)
29. Van Craenendonck, T., Blockeel, H.: Using internal validity measures to compare clustering algorithms. Benelearn 2015 Poster presentations (online) pp. 1–8 (2015)
30. Wang, K., Zhang, J., Li, D., Zhang, X., Guo, T.: Adaptive affinity propagation clustering. arXiv preprint arXiv:0805.1096 (2008)
31. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 52–67 (2018)
32. Wu, J., Wang, Y., Xue, T., Sun, X., Freeman, B., Tenenbaum, J.: Marrnet: 3d shape reconstruction via 2.5 d sketches. In: Advances in neural information processing systems. pp. 540–550 (2017)
33. Wu, J., Zhang, C., Xue, T., Freeman, B., Tenenbaum, J.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: Advances in neural information processing systems. pp. 82–90 (2016)
34. Wu, J., Zhang, C., Zhang, X., Zhang, Z., Freeman, W.T., Tenenbaum, J.B.: Learning shape priors for single-view 3d completion and reconstruction. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 646–662 (2018)

35. Yan, X., Yang, J., Yumer, E., Guo, Y., Lee, H.: Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In: Advances in neural information processing systems. pp. 1696–1704 (2016)
36. Yang, B., Wen, H., Wang, S., Clark, R., Markham, A., Trigoni, N.: 3d object reconstruction from a single depth view with adversarial learning. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 679–688 (2017)
37. Yang, G., Huang, X., Hao, Z., Liu, M.Y., Belongie, S., Hariharan, B.: Pointflow: 3d point cloud generation with continuous normalizing flows. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4541–4550 (2019)
38. Yang, Y., Feng, C., Shen, Y., Tian, D.: Foldingnet: Point cloud auto-encoder via deep grid deformation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 206–215 (2018)