Ensemble AutoDecoder CompNets for 3D Point Cloud Pair Encoding and Classification

Samridha Shrestha
Computer Science, NYUAD
sms1198@nyu.edu

Advised by: Yi Fang

ABSTRACT

Since the inception of LeNet5, Yan LeCun's first Convolutional Neural Network for handwritten digit-recognition in 1998, Neural Networks have come a long way in classifying 2D images and extracting their features. However, since we live in a 3D world, it is natural to ask if we can classify 3D shapes and extract their shape descriptors. This turns out to be rather difficult given the exponential increase in complexity with the added third dimension. To perform this classification efficiently, we propose an ensembled encoderless-decoder and classifier neural network to learn the latent encoding vectors for 3D point-cloud pair transformations. This ensemble extracts the feature descriptors of the transformations without a parametric encoder or a lossy 2D projection. We also empirically demonstrate the effectiveness of our ensemble model through a benchmark comparison on a 3D point cloud classification task for the PointNet and ModelNet dataset. Finally, we discuss the applicability of our 3D shape descriptor learners and classifiers as preprocessing techniques for eventual 3D object classification, segmentation, and database retrieval tasks.

KEYWORDS

AutoDecoder, 3D Point Cloud, 3D object classification, Ensemble Neural Networks, Encoderless-decoders, latent point drift

Reference Format:
Samridha Shrestha. 2020. Ensemble AutoDecoder CompNets for 3D Point Cloud Pair Encoding and Classification. In *NYUAD Capstone Seminar Reports, Spring 2020, Abu Dhabi, UAE.* 11 pages.

جامعة نيويورك أبوظبي
NYU ABU DHABI

Capstone Seminar, Spring 2020, Abu Dhabi, UAE

1 INTRODUCTION AND BACKGROUND

Over the last few decades, Convolutional Neural Networks (CNNs) have revolutionized the field of computer vision. First used by the US Postal Service to classify zip codes in the 1990s [20], CNNs have progressed in other sub-domains of computer vision such as latent encoding and feature extraction of 2D images as vectors. These encodings have been successfully used to create image segmentation CNNs [3] that use encoders to generate the compressed latent representation of the image which are upsampled to regionally segmented images using a decoder. The latent representations of images can also be utilized to create generative models such as variational autoencoders [26] that can output new images drawn from the latent distribution of the original image. Other formulations of this latent space, such as random noise vectors are also used for Generative Adversarial Networks (GAN) [12] to create new images. The latent encodings represent a compact representation of the image features required by many computer vision models. Machine learning models cannot understand the raw format of images so the features of the images must be extracted before image segmentation or retrieval tasks. [25] have developed a content-based image retrieval system that uses Haar wavelet filters to extract binary textures from images that are fed to a Support Vector Machine for supervised retrieval learning.

Clearly, the latent representations of images are significant to 2D image segmentation, retrieval, and labeling tasks. Deep CNNs have been particularly successful at extracting these 2D image latent features that cannot be handcrafted. The next logical step is to investigate the applicability of such feature extraction to the 3D world. 3D shape models have become more accurate with laser scanners and become more popular in the last decade with the emergence of 3D model search engines [21]. These models have become a mainstay in mechanical engineering, industrial material design, molecular biology, drug manufacturing, and virtual-reality [28]. However, for the 3D shapes to be useful in these domains, computer vision models require shape features or descriptors from 3D objects just like their 2D counterparts. These models

require that descriptors efficiently perform tasks like point-registration, shape reconstruction, 3D segmentation, classification, and database retrieval. The 3D shape descriptors are n-dimensional vectors or encodings serving as compressed representations of the shapes. These encodings contain vectorized global and local feature descriptors of the shape. The global features contain information about the overall geometry or shape of the 3D object while the local features describe the surface normals, curvatures, and the local geometry of the shape segments [13]. The local features are useful for point set registration or matching of one shape to another, while global features aid object recognition and shape retrieval. These compact features stored as vectors can be used for database querying of 3D shapes whereas large volumes of 3D object models can be stored as compressed vectors.

To understand the 3D shape descriptor extraction process, we have to first internalize the different paradigms for 3D shape learning. The first is mesh-based representations, where stitches of triangles or quadrilaterals make up the 3D body with predefined templates. This mesh-based method provides detailed rendering of real-world shapes but only shapes with fixed mesh topology can be modeled [4].

Another natural extension for 3D shape learning from the 2D pixel-based world is the 3D voxel-based representation. Voxels are 3D grids that volumetrically define a 3D body with an occupancy grid that are non-parametric (data-driven and not represented by any mathematical formulation). However, voxel representations are information sparse and have increased compute requirements that grow cubically (due to the added z-dimension) instead of quadratically, as in the 2D counterpart. Therefore, this voxel representation cannot reproduce life-like objects with the same fidelity as mesh-based representations and are limited to resolutions of 128^3 voxels [24]. Even methods that reduce the cubic complexity requirements, like octree voxel representations, are only limited to a resolution of 512^3 [31].

The final, more popular representation of 3D shapes is point-clouds. Point clouds are non-parametric, potentially sparse representations of 3D objects where individual points store the x, y, and z coordinate information. Such representations match directly with the outputs generated by laser scanners, light detection and ranging (LiDAR) cameras. Hence, the point-cloud approach is directly applicable to 3D learning in the real world. Point-clouds, however, cannot represent the underlying continuous 3D geometry and the topology of an object compared to CAD or simple mesh models. Nonetheless, point clouds save computation and space requirements compared to mesh-presentations as combinatorial connectivity patterns and multiple primitive encodings do not have to be learned [11]. Therefore, point-clouds allow efficient manipulations for geometric transformations or deformations [11]. For these reasons, we chose the point-cloud data format for our research as well.

With these representations of 3D objects in mind, we will now look at different methods used to extract the 3D shape features required for 3D object recognition, segmentation, point-registration, and database-retrieval pipelines. [34] have proposed a 3D counterpart of the Generative Adversarial (GAN) setup to learn the probabilistic latent space of 3D shapes using a voxel representation. However, this GAN setup is known to be notoriously difficult to train and stabilize [24]. Recently, discriminator-less GANs have been proposed which overcome this stability issue by removing the discriminator completely [7]. Beyond GANs, autoencoders have also been used in the 3D domain like just in the 2D domain. [10] use voxel-based 3D convolution layers to infer the features of partial 3D shapes and generate the complete high-resolution 3D shape using a bottleneck encoding vector. [24], on the other hand, uses a completely different network of encoder-less decoder networks or AutoDecoders. Since the encoders are not used during the testing phase in encoder-decoder networks, [24] discards the encoder completely and relies on a decoder alone to learn the optimal latent shape descriptor of a 3D shape. Other formulations to learn the latent shape descriptors of 3D shapes include volumetric and multi-view CNNs, where the 3D object is convolved using a 2D CNN from different angles or views to extract the shape descriptors with good accuracy, but at increased computational costs [27]. It is evident that the increased complexity of working with 3D models requires novel feature extraction methods that so many 3D object classification, segmentation, and database retrieval pipelines require.

2 OUR IDEA AND CONTRIBUTIONS

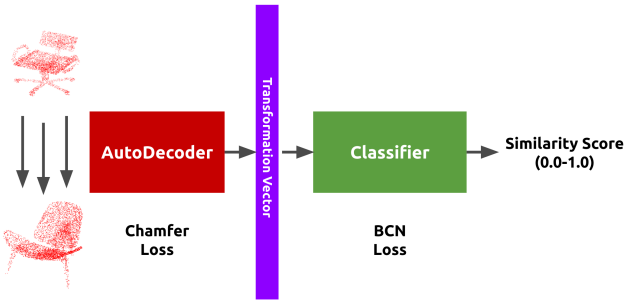


Figure 1: Ensemble AutoDecoder-CompNet training process

We propose an ensemble encoder-less decoder comparison network (Ensemble ADNet-CompNet) to extract the 3D shape

descriptors or the latent vectors that define the geometric transformation between two point clouds for 3D object classification and database retrieval. Our ensemble network is inspired by the autodecoder paradigm from [24] and the point-cloud pair latent vector extraction formulation from EPD-Net [36]. Since the encoder is never used after the training process, we decided to completely forgo the encoder and use an ensemble of AutoDecoders. This ensemble extracts the latent vector formulation through a self-supervised reconstruction task optimized by reducing the chamfer loss between the source and target point clouds.

Our network pipeline consists of two major neural network ensembles: the autodecoder and the classifier. The autodecoder learns the point set registration or the spatial transformation from one 3D point-cloud to another through the coherent point drift task. The coherent point drift task is approached as a maximum likelihood estimation problem over the continuous displacement field that aligns a source point cloud to a target point cloud. [22, 32] [36] The autodecoder encapsulates this point drift in a latent encoding vector that represents the continuous geometric transformation from one shape to another. These latent vector representations optimized by the autodecoder are robust to rigid (translation, rotation, or reflection) and non-rigid transformations (shearing and scaling) and are highly generalizable for other point-cloud applications. Additionally, our method is superior to one of the most popular methods for point registration, the Iterative Closest Point algorithm, since the algorithm cannot function with non-rigid transformations. [6].

The classifier is an ensemble model of shallow neural networks that ingest the encoding vector optimized by the autodecoder as input and output a similarity score. The score empirically states the class similarity of the two-point clouds, which is then used for 3D object classification. The classifier ensemble avoids overfitting and reduces individual model variance. We empirically demonstrate this reduction in generalization error and the increase in accuracy for 3D object classification in the PointNet and ModelNet datasets as compared to the single network autodecoder proposed in the EPD-NeT [36].

Our network is a stacked ensemble that is different from the traditional bagging or boosting models used in ensemble learning. Our classifiers are either initialized with different weights or have structures of varying depths to induce model independence. The encoding vector is fed to each classifier and the outputs are combined by another fully-connected neural network instead of the averaging, hard, or soft-voting approaches that were used in [19] for 3D object classification.

Ensemble models have also been proven to perform well in computer vision classification tasks. Google's GoogleNet ensemble module was one of the top entries in the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) image classification task [29]. While [35] used horizontal and vertical ensemble models for the ICML 2013 image classification Black Box challenge to land in third place in the public leaderboard. Image classification using ensemble algorithms with deep learning and hand-crafted features has also been demonstrated to increase state-of-the-art classification accuracy in the CIFAR-10 dataset [30]. Consequently, it is natural to assume ensemble models will generalize well to 3D classification tasks.

In summary, our contributions are:

(1) An Encoder-less ensemble approach for retrieval of both local and global level 3D shape descriptors between point-cloud pairs representing their geometric transformation relationship.
(2) 3D point-cloud pair latent encoding extraction method that can be used for efficient 3D object database query and retrieval based on transformations.
(3) The whole ADNet-CompNet ensemble provides a 3D object classification pipeline. The model learns relatively fast, overfits less, and generalizes well compared to the non-ensemble model. We empirically demonstrate this based on accuracy, f1-score, and Receiver Operating Characteristic Area Under Curve (ROC-AUC) metrics on 3D shape classification tasks on the PointNet, and ModelNet datasets.
(4) Ensembles that scale well to small and large datasets given the complexity of our 3D point cloud data and relative training times on our system.

Our full code is also publicly available at our GitHub repository.

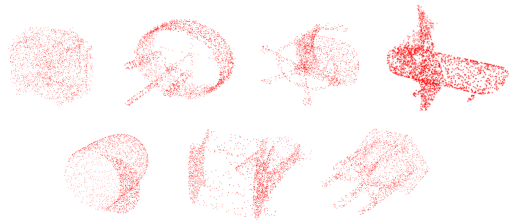
3 METHODOLOGY

3.1 Data

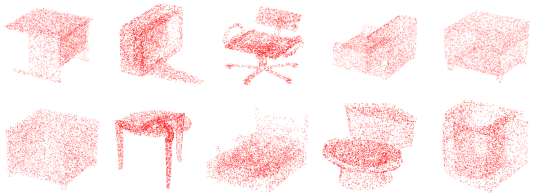
We use the PointNet7 and PointNet Full [9] and the ModelNet data [38] for training and testing our ADNet-CompNET ensemble models. The PointNet7 and PointNet Full contain 10817-2507 and 12137-2874 train-test point-clouds with 2048 points in each cloud. ModelNet10 and ModelNet40 have 3991-908 and 7598-1860 train-test point-clouds respectively, with each cloud having 5000 points. All point-clouds were segmented and contained a single 3D object from a distinct class. However, we could only get 33 out of the 40 classes in the ModelNet40 dataset because our .off to .npy conversion module (kaolin [15]) could not resolve some of the .off files in the ModelNet40 dataset.

Data preprocessing only involved converting all .off mesh files into a .npy NumPy format point cloud file that could be loaded as tensors into our models.

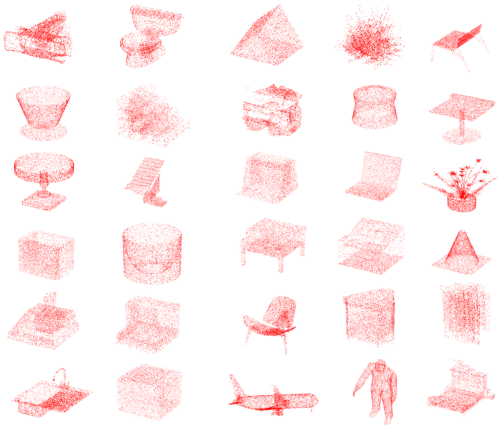
The distribution of the ModelNet10 and ModelNet40 training samples for each class are listed in the supplementary material.



(a) PointNet7 Point Cloud Samples



(b) ModelNet10 Point Cloud Samples



(c) ModelNet40 Point Cloud Samples

Figure 2: Point Cloud samples from training and testing data

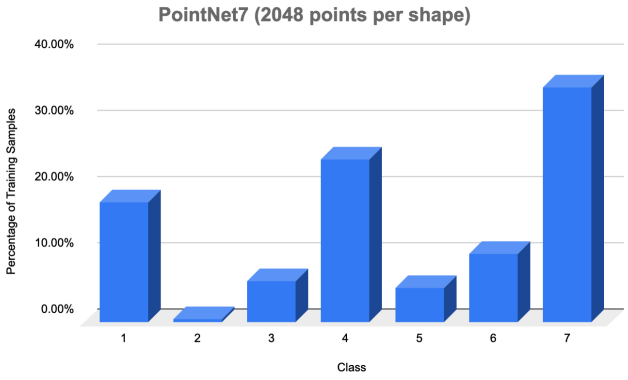


Figure 3: PointNet7 Training Data Class distribution

3.2 Coherent Point Drift Task between 3D Point Clouds

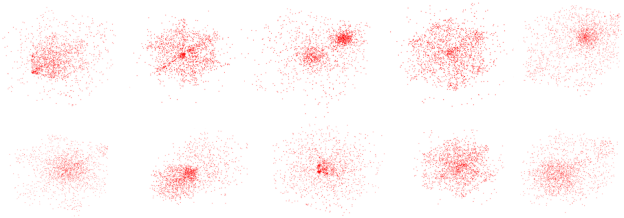


Figure 4: Point Drift representation between point-clouds of different classes

The coherent point-drift task is done by a complex continuous function to calculate the optimal point-set registration encoding. This continuous displacement function can be represented as $\tau = f(X) : X_i \text{ -> } Y_i, i = 1, ...n$. The function displaces all points in the source point cloud X to the target cloud Y. A latent encoding vector z_{XY} represents this transformation. The AutoDecoder combines the classical encoder $q_\theta(z|x)$ and the decoder $p_\phi(y|z)$ networks into one function, τ.

The optimization function we use is the chamfer distance loss with the Euclidean metric. The chamfer loss is a suitable metric for empirically measuring the similarity between the source and target point cloud. The Chamfer loss can be calculated efficiently and the function is differentiable with respect to point locations for backpropagation. In addition, chamfer distance loss is also robust to the small number of outlier points that might be present in the point-cloud sets. Other distance metrics like Hausdorff distance were not used since it does not tolerate a small number of outliers in the point sets. Chamfer distance also does not require a total bijection between points in the point cloud pairs which is required for calculating the Earth Mover's Distance. [11]

In essence, the AutoDecoder optimizes the following formulation of the latent vector z_{XY} that is also proposed in the EPD-Net implementation [36]

$$z_{X,Y} = \arg\min_{z} \; L_C(\tau(z, X), Y)$$

The complex continuous displacement function $\tau(z, X)$ in this case is the ensemble AutoDecoder network that calculates the coherent point drift between the two-point clouds X and Y to create a representative vector z_{XY} to encode the geometric transformation . [32, 36] The AutoDecoder network can only process preprocessed and segmented individual point clouds which is true for all the datasets that we use [9, 38].

3.3 AutoDecoder Network (ADNet)

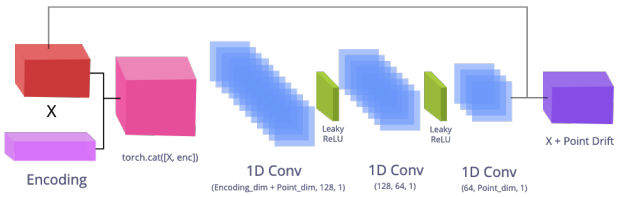


Figure 5: Single AutoDecoder Network

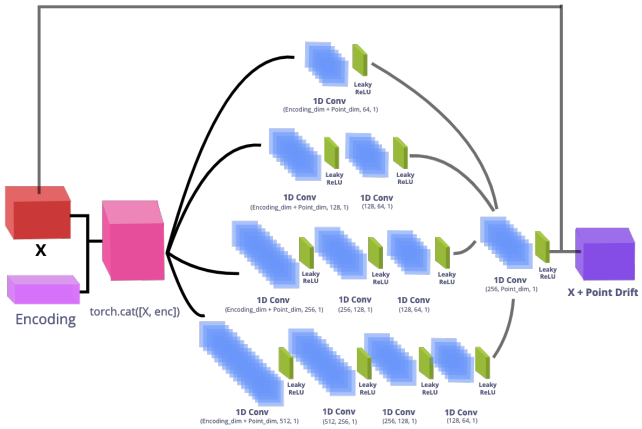


Figure 6: Ensemble AutoDecoder Network

The latent vector z is an embedding layer with a dimension of 256 values. The vector's weights are initialized with a normal distribution of mean 0 and standard deviation of $\frac{1}{\sqrt{256}}$ which prevents exploding or diminishing values in the vector after back propagation.

These latent vectors are concatenated to each point cloud samples and fed into each of the AutoDecoders in the ensemble (fig 6). The inputs are convolved with multiple 1D Convolutional layers paired with leaky ReLU activations. Each section in the ensemble represents a network with increasing depths to capture both local level features and global level features and more importantly to induce independence among the models. The resulting displacements are concatenated and fed into another 1D Convolutional layer with leaky ReLU to output a displacement that is added to the original point cloud data X to get a transformed point cloud X'. We use 1D convolutions as its computational complexity is significantly lower given that it only involves array operations instead of matrix operations that are typical in classical multidimensional convolutions. Research also shows that 1D convolutions optimize efficiently and train faster with shallower networks [18]. The AutoDecoder then uses ADAM [17] to iteratively optimize its weights and the latent vector using the backpropagation algorithm [20] minimizing the Chamfer reconstruction loss. Data loaders from PyTorch create batches of the same class and different class point-cloud pairs from the 3D point–cloud datasets that are used as inputs to the AutoDecoder Network.

3.4 Comparison Network (CompNet)

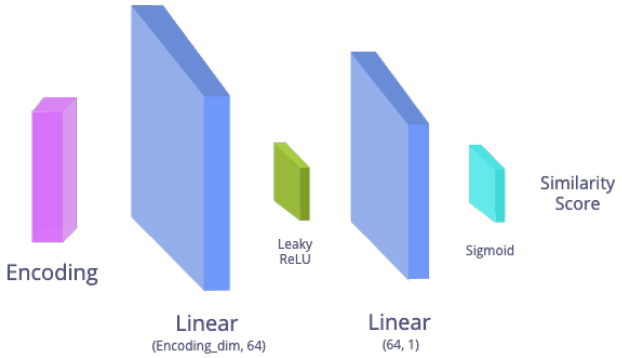


Figure 7: Single CompNet Network

The Comparison network (CompNet) is a shallow neural network with only one fully connected hidden layer. The CompNet takes the latent encoding generated by the AutoDecoder as input and generates a class similarity score at the end of its feed-forward architecture using the sigmoid activation.

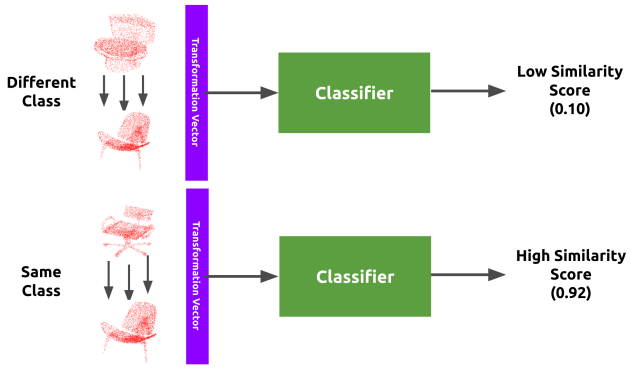
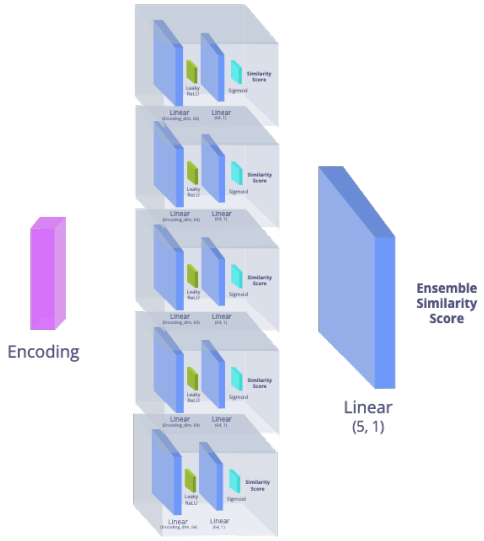
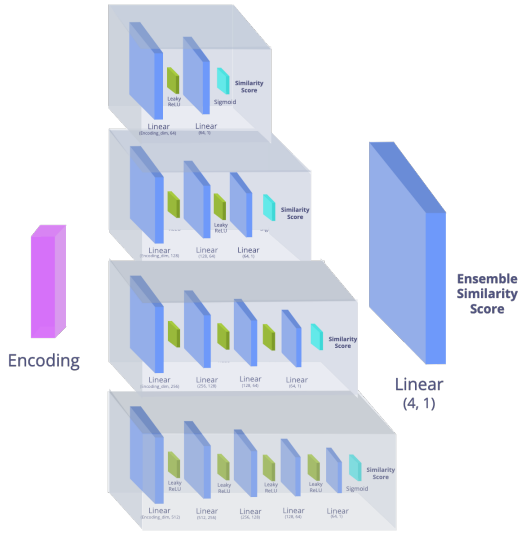


Figure 9: Full encoding extraction and classification pipeline

During the training process, the network is presented with labeled encoding vectors that represent transformations between the same or different classes. These labeled training sets are used by the CompNet to optimize its hidden layer



(a) Ensemble CompNet Architecture 1



(b) Ensemble CompNet Architecture 2

Figure 8: Ensemble CompNet Architectures

weights and increase its binary classification accuracy. CompNet also uses the ADAM optimizer with backpropagation to increase its classification accuracy by reducing the binary cross-entropy loss associated with the classification task. We use two different architectures for the CompNet ensembles: one is similar to the ADNet ensemble with networks of varying depths (fig 8a) but the other architecture (fig 8b) uses multiple identical shallow CompNets that achieve independence through random seeding initialization of their layer weights.

In our CompNet ensemble, the encoding vector is fed to each CompNet that generates its similarity score which is then fed into another fully connected layer that combines the outputs of individual CompNets. The final output aggregation layer is also a part of the entire backpropagation network. Such Stacked ensemble formulation achieves a sophisticated form of cross-validation which is not always feasible with neural networks trained on large datasets [33]. This network stacking greatly reduces the generalization error of the whole network as observed in our results.

After some hyper-parameter tuning, we observed shallow CompNets performed better in the classification task over deep networks with l2 regularization or dropout layers. All of the other hyperparameters used for training our models are provided in the supplementary material.

4 SYSTEM SPECIFICATIONS FOR TRAINING AND TESTING

We use a system with the following specifications for training and testing our models:

- CPU: Intel(R) Core(TM) i9-9900K CPU @ 3.60GHz
- Architecture: 64 Bit
- RAM: 62 Gib System Memory
- Graphics: 7979 MiB GeForce RTX 2080

Detailed specifications of our system are also available in the supplementary material.

5 RESULTS AND DISCUSSION

Model	Total Accuracy	Same Class F1-score	Different Class F1-score	ROC AUC
Base CompNet	*0.87*	*0.87*	*0.87*	*0.943*
SVM	0.80	0.81	0.79	NAN
Decision Trees	0.74	0.73	0.74	0.810
Logistic Regression	0.80	0.80	0.80	0.882
Random Forest	0.79	0.79	0.79	0.868
Naive Bayes	0.78	0.78	0.77	0.850
Ensemble CompNet1	**0.97**	**0.97**	**0.97**	**0.998**
Ensemble CompNet2	**0.96**	**0.96**	**0.96**	**0.995**

Table 1: Metrics for the classification task on the PointNet7 dataset

Model	Total Accuracy	AutoDecoder Train Time	Classifier Train Time
Base CompNet	0.87	5m 17s	25.2s
SVM	0.80	5m 17s	1.03s
Decision Trees	0.74	5m 17s	1.30s
Logistic Regression	0.80	5m 17s	993ms
Random Forest	0.79	5m 17s	9.79s
Naive Bayes	0.78	5m 17s	869ms
Ensemble CompNet1	0.97	5m 17s	14.3s
Ensemble CompNet2	0.96	5m 17s	18.5s

Table 2: Accuracy and train time for the classification task on the PointNet7 dataset

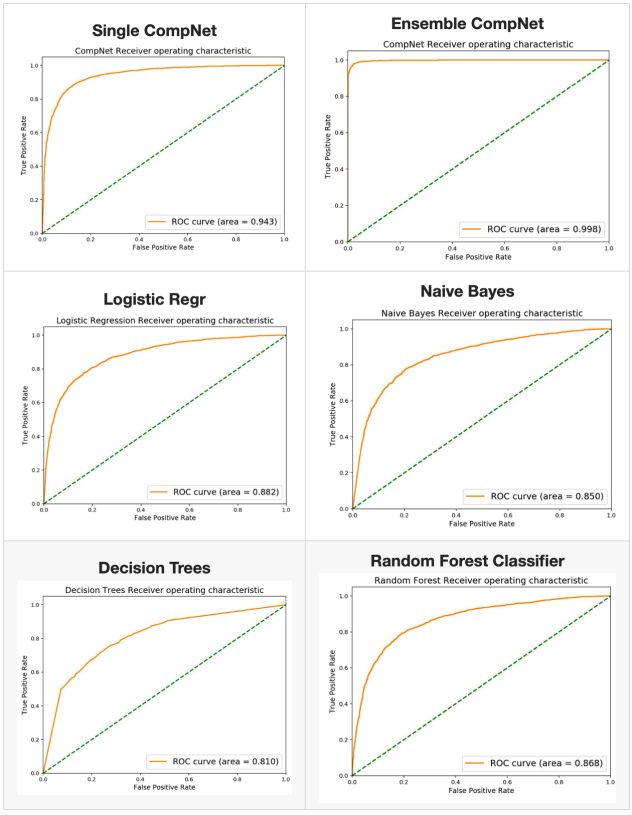


Figure 10: ROC-AUC benchmark comparison between all classifier models

The results of the latent encoding learning and class similarity classification task immediately show that the ensemble CompNet significantly outperforms the single CompNet architecture (EPD-Net architecture) boosting accuracy from 87% to 97%. This improvement is even more evident when we train our ADNet-CompNets on the full PointNet data as ensemble CompNets accuracy is increased to an average of 98% from 87% on the single CompNet. EPD-Net uses a deeper architecture similar to the single CompNet structure but is prone to overfitting to the training data, which would explain its lower accuracy on the classification task. [36]

After training the AutoDecoder on 10817 samples from PointNet7, the ensemble CompNets architectures beat all other models in terms of classification accuracy by 10% on average (Table 1). The accuracies of the model themselves had an average error of ± 1% as accuracy fluctuates by ± 1% during repetitions. We also show that our neural network models (Single and Ensemble CompNets) always outperform all other classical supervised machine learning models in terms of accuracy, f1-scores and Receiving Operator Characteristic Area Under Curve (ROC-AUC) for the 3D object classification task (Table 1, figure 10). These results demonstrate that neural networks are highly suited to the task of 3D object classification using latent transformation encodings compared to non-neural network models.

The training times for the PointNet7's 10817 point-cloud set show that our method computes latent encodings on an average of 5 minutes 15 seconds and similarity scores within 20 seconds. 2 These temporal metrics suggest that our models are scalable to larger datasets as well.

However, when we test our models by introducing ensemble AutoDecoders along with ensemble CompNets, we only observe slight (Table 4) or no improvement (Table 5) in the test set accuracy on the Modelnet10 or the Modelnet40 datasets. Interestingly, the performance for Ensemble AutoDecoder Nets actually decreased for the PointNet7 classification (Table 3).

The explanation for this reduced performance might be caused by the fact that point-set registration tasks require local feature extraction which is more effective with shallower networks. Our ensemble network outputs might have been overwhelmed by the deeper networks that actually capture the global level features which are not suitable for point-set registration tasks [28]. And the 1D convolutions we use in the AutoDecoders perform much better with shallower neural networks compared to deeper ones as well [18]. Another reason for the reduced accuracy of the Ensemble ADNet could be the highly imbalanced nature of the training set for the 'PointNet7' dataset where some classes only represent 0.36% of the total number of training samples. 3

In summary, regardless of the AutoDecoder model, all empirical results do support our hypothesis that ensemble

AutoDecoder	Classifier	Total Accuracy	Same Class F1 Score	Diff Class F1 Score	ROC AUC
Single AutoDecoder	Single CompNet	0.87	0.87	0.87	0.941
	Ensemble CompNet	0.98	0.98	0.98	0.998
Ensemble AutoDecoder	Single CompNet	0.81	0.81	0.82	0.896
	Ensemble CompNet	0.95	0.95	0.95	0.989

Table 3: Performance of Single and Ensemble AutoDecoder on PointNet7

AutoDecoder	Classifier	Total Accuracy	Same Class F1 Score	Diff Class F1 Score	ROC AUC
Single AutoDecoder	Single CompNet	0.63	0.69	0.54	0.688
	Ensemble CompNet	0.77	0.78	0.75	0.864
Ensemble AutoDecoder	Single CompNet	0.64	0.62	0.67	0.724
	Ensemble CompNet	0.80	0.78	0.80	0.890

Table 4: Performance of Single and Ensemble AutoDecoder on ModelNet10

AutoDecoder	Classifier	Total Accuracy	Same Class F1 Score	Diff Class F1 Score	ROC AUC
Single AutoDecoder	Single CompNet	0.59	0.59	0.59	0.637
	Ensemble CompNet	0.73	0.70	0.76	0.817
Ensemble AutoDecoder	Single CompNet	0.59	0.55	0.62	0.625
	Ensemble CompNet	0.73	0.70	0.76	0.817

Table 5: Performance of Single and Ensemble AutoDecoder on ModelNet40

CompNets do exceptionally well in the class-similarity classification task compared to single neural-network architectures. Therefore our model shows great promise in extracting the 3D latent features of point-cloud transformations that have great applications in many 3D shape learning domains.

6 SUMMARY AND IMPACT

We have demonstrated the efficacy of our ensemble ADNet-CompNet formulation to learn a compact geometric transformation encoding between point-cloud pairs through the coherent point drift task and generate shape similarity scores. We empirically confirmed the classification superiority of our ensemble compared to other non-ensemble models by

benchmarking on the PointNet and ModelNet datasets. (Table **??**)

Our results show our novel ensemble neural network formulation can extract the 3D shape descriptors of the point-cloud point drift as representative latent vectors. These vectors can be used for 3D point-cloud database retrieval tasks where the latent vectors are used for querying into the database instead of the entire point clouds. Our model can also be used for effective 3D shape classification by using the encodings as demonstrated. The ADNet-CompNet model can be used to generate geometric transformations for 3D scenes based on data generated by LiDARs and depth cameras which do not contain the full 3D surface information. This capability to generate such transformations will allow 3D computer vision in robots to properly understand the relationship between 3D objects of the same and different classes in their environment. In addition such latent representations of the point-set registration between point-cloud pairs have enormous applications in autonomous driving [37], robotic manipulation, [8] 3D image stitching, medical imaging, and virtual augmented reality [23].

The 3D model domain will only grow in the future. Nvidia recently announced NVIDIA Omniverse, an open design collaboration platform to share modeling, animation, lighting, visual effects, and rendering information on 2D and 3D models across multiple applications. [1] This repository is a prime candidate for implementing a 3D shape retrieval system based on the latent encodings generated by our model.

In computer graphics, the available computing capability has overtaken the Input-Output capacity of computers, and scientists have to significantly limit the amount of data in persistent storage when a graphics simulation runs [14]. Our framework could be used for in-situ visualizations of transformations from one object to another since the transformation is stored as compact latent vectors which can generate visualizations in-situ without writing to persistent storage. This in-situ simulation visualization can be done using the NVIDIA Omniverse pipelines as demonstrated in [14].

However, perhaps one of the largest areas where 3D model design and encoding extraction can make an impact is in drug design. In response to the COVID-19 pandemic, Ford's Apollo Project designed and manufactured air-ventilation units as personal protective equipment (PPE) for medical workers [2]. The PPE unit was production-ready in 38 days, from conception as 3D models to working prototypes. Our autodecoder ensemble model could be used to extract the latent encoding transformations between 3D constructions of human heads to create enclosed ventilators that suitably fit the average human. This is just one example of the kind of powerful impact 3D model-design technology can have that goes beyond computer graphics.

7 RELATED WORK

For the task of extracting 3D feature descriptors using latent encodings, [24] propose a signed distance function (SDF) deep neural network model. Our ADNet is inspired by their SDF autodecoder network. Their SDF is a function that represents the continuous volumetric field: negative values represent regions inside the shape, positive values represent regions outside, and zero-values represent the shape boundary to high fidelity. Their models achieve state-of-the-art 3D shape representation and completion learning [24]. However, the SDF models work on the assumption that a complete view of the model's surface is available as ground truth for training, which is not always guaranteed for data acquired from LiDARs and depth cameras. 3D shape nets [24]

3D ShapeNets approach this 3D feature extraction problem through the use of Convolutional Deep Belief Networks on 3D voxel grids that are represented as probability distributions of binary variables [38]. The ShapeNet learns the 3D shape descriptors through a hierarchical part representation of the data and hallucinates the missing sections. The probabilistic approach suffers from the voxel-based representation problem where resolutions above 1283 cannot be handled by current computers [24].

Other 3D encoding extraction methods include the use of multi-view networks. As opposed to volumetric CNNs, multi-view CNNs use traditional CNNs on different 2D views of the 3D model to learn the shape descriptors of the 3D object. [27] show that the multi-view models can significantly outperform volumetric CNNs on object classification and retrieval tasks. Their multi-view model projects 3D objects to a 2D space using a method similar to Computed X-ray scans instead of an external snapshot from multiple directions. [27] achieve this projection method using an anisotropic convolutional kernel that creates different views by convolving through the shape in different directions. Another multi-view 3D shape extraction and classification architecture, rotation net, has a state-of-the-art accuracy of 98% on the ModelNet40 3D object classification task [16]. Rotation net jointly estimates the pose or orientation of the 3D shapes and their categories through optimization of the initially unknown viewpoint variables as latent values. RotationNet takes an unsupervised learning approach that avoids training bias that might occur with other multi-view techniques that ingest labeled and fixed view-points [16, 27]. However, few ensemble approaches have been proposed for the extraction of 3D shape descriptors from the recognition and retrieval of 3D shapes. [19] propose a point-cloud based ensemble deep learning model that uses a different ensemble architecture compared to our method. The ensemble includes pre-trained existing architectures like PointNet, DeepSets, PointCNN, and KCNet. To independently train these ensembles, they

experiment with bagging methods that randomly sample cloud sets from the total set and compare results from hard voting, soft voting, and direct output averaging. They also show that two models with significantly disparate architectures achieve a 94.03% accuracy on the ModelNet40 dataset with results comparable to those produced by 10 models of the same architecture [19].

8 FUTURE WORK

The next logical step to our work is to train our ensemble autodecoder CompNet model on larger datasets such as the ShapeNet dataset [38], as one of the problems we encountered during training was the unbalanced sample size for some of the classes in the PointNet dataset. Training with a well-balanced or larger dataset for all classes will yield improved accuracies for all object classes. Oversampling can also be used to generate synthetic point-clouds similar to the under-represented class, with random point generation using the Edited K-Nearest Neighbors as in SMOTE ENN [5].

Another significant improvement to our model would be to create a more efficient computation pipeline for the full classification task. Currently, our model can efficiently compute whether two 3D point clouds are of the same or different classes. However, without a priori knowledge of the point-cloud classes, classification of the point-cloud from a different class would require a pairwise comparison algorithm that has a quadratic (n2) runtime. A 3D point-cloud pair transformation encoding database retrieval algorithm must be more computationally efficient for this classification task.

Additionally, regarding ensemble architectures, [19] report that two neural network ensembles with different structures performed equally well compared to 10 ensemble nets with the same structure in the ModelNet40 classification task. This computation improving setup could be exploited by modifying our autodecoder network to include a few ensembles of vastly different structures instead of increasing the depths of 1D convolutions (Fig. 6). The number of learners to use in our ensemble and their weight initialization algorithms are hyperparameter tuning problems. In effect, the issue of choosing parameters for such ensemble models for 3D latent encoding extraction is a valuable future research topic.

REFERENCES

[1] March 20, 2019. NVIDIA introduces Omniverse. *Datamonitor* (March 20, 2019).

[2] Maria Aspan. 2020. Ford Shifts Gears. *Fortune* 181, 5 (2020), 22 – 24. http://proxy.library.nyu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=142757530&site=eds-live

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (2017), 2481–2495.

[4] T. Bagautdinov, C. Wu, J. Saragih, P. Fua, and Y. Sheikh. 2018. Modeling Facial Geometry Using Compositional VAEs. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3877–3886.

[5] Gustavo Batista, Ronaldo Prati, and Maria-Carolina Monard. 2004. A Study of the Behavior of Several Methods for Balancing machine Learning Training Data. *SIGKDD Explorations* 6 (06 2004), 20–29. https://doi.org/10.1145/1007730.1007735

[6] P. J. Besl and N. D. McKay. 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (1992), 239–256.

[7] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. 2019. Optimizing the Latent Space of Generative Networks. *arXiv:1707.05776 [cs, stat]* (May 2019). http://arxiv.org/abs/1707.05776 arXiv: 1707.05776.

[8] Berk Calli, Arjun Singh, James Bruce, Aaron Walsman, Kurt Konolige, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. 2017. Yale-CMU-Berkeley dataset for robotic manipulation research. *The International Journal of Robotics Research* 36, 3 (2017), 261–268. https://doi.org/10.1177/0278364917700714 arXiv:https://doi.org/10.1177/0278364917700714

[9] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Honolulu, HI, 77–85. https://doi.org/10.1109/CVPR.2017.16

[10] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. 2017. Shape Completion Using 3D-Encoder-Predictor CNNs and Shape Synthesis. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 6545–6554.

[11] Haoqiang Fan, Hao Su, and Leonidas Guibas. 2016. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. *arXiv:1612.00603 [cs]* (Dec. 2016). http://arxiv.org/abs/1612.00603 arXiv: 1612.00603.

[12] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. *ArXiv* abs/1406.2661 (2014).

[13] Xian-Feng Han, Jesse S. Jin, Juan Xie, Ming-Jie Wang, and Wei Jiang. 2018. A comprehensive review of 3D point cloud descriptors. *ArXiv* abs/1802.02297 (2018).

[14] Mathias Hummel and Kees van Kooten. 2019. Leveraging NVIDIA Omniverse for In Situ Visualization. In *High Performance Computing*, Michèle Weiland, Guido Juckeland, Sadaf Alam, and Heike Jagode (Eds.). Vol. 11887. Springer International Publishing, Cham, 634–642. https://doi.org/10.1007/978-3-030-34356-9_48

[15] Krishna Murthy J., Edward Smith, Jean-Francois Lafleche, Clement Fuji Tsang, Artem Rozantsev, Wenzheng Chen, Tommy Xiang, Rev Lebaredian, and Sanja Fidler. 2019. Kaolin: A PyTorch Library for Accelerating 3D Deep Learning Research. *arXiv:1911.05063* (2019).

[16] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. 2016. RotationNet: Joint Learning of Object Classification and Viewpoint Estimation using Unaligned 3D Object Dataset.

[17] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2015).

[18] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman. 2019. 1D Convolutional Neural Networks and Applications: A Survey. *arXiv:1905.03554 [cs, eess]* (May 2019). http://arxiv.org/abs/1905.03554 arXiv: 1905.03554.

[19] Daniel Koguciuk, Łukasz Chechliński, and Tarek El-Gaaly. 2019. 3D Object Recognition with Ensemble Learning — A Study of Point Cloud-Based Deep Learning Models. *arXiv:1904.08159 [cs]* (May 2019). http:

//arxiv.org/abs/1904.08159 arXiv: 1904.08159.

[20] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. 1989. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation* 1, 4 (1989), 541–551. https://doi.org/10.1162/neco.1989.1.4.541 arXiv:https://doi.org/10.1162/neco.1989.1.4.541

[21] Patrick Min, John A. Halderman, Michael Kazhdan, and Thomas A. Funkhouser. 2003. Early Experiences with a 3D Model Search Engine. In *Proceedings of the Eighth International Conference on 3D Web Technology* (Saint Malo, France) *(Web3D '03)*. Association for Computing Machinery, New York, NY, USA, 7–ff. https://doi.org/10.1145/636593.636595

[22] Andriy Myronenko, Xubo Song, and Miguel Carreira-Perpiñán. 2006. Non-rigid point set registration: Coherent Point Drift. *NIPS* 19, 1009–1016.

[23] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. 2011. KinectFusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. 127–136.

[24] Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. 2019. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 165–174.

[25] Palepu Pavani and T. Sashi Prabha. 2014. Content Based Image Retrieval Using Machine Learning Approach. In *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2013*, Suresh Chandra Satapathy, Siba K Udgata, and Bhabendra Narayan Biswal (Eds.). Vol. 247. Springer International Publishing, Cham, 173–179. https://doi.org/10.1007/978-3-319-02931-3_21

[26] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. 2016. Variational Autoencoder for Deep Learning of Images, Labels and Captions. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 2352–2360. http://papers.nips.cc/paper/6528-variational-autoencoder-for-deep-learning-of-images-labels-and-captions.pdf

[27] Charles R. Qi, Hao Su, Matthias NieBner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. 2016. Volumetric and Multi-view CNNs for Object Classification on 3D Data. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Las Vegas, NV, USA, 5648–5656. https://doi.org/10.1109/CVPR.2016.609

[28] R. Rostami, F. S. Bashiri, B. Rostami, and Z. Yu. 2019. A Survey on Data-Driven 3D Shape Descriptors. *Computer Graphics Forum* 38, 1 (Feb. 2019), 356–393. https://doi.org/10.1111/cgf.13536

[29] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), 1–9.

[30] Erdal TaSci and Aybars Ugur. 2018. Image classification using ensemble algorithms with deep learning and hand-crafted features. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*. IEEE, Izmir, Turkey, 1–4. https://doi.org/10.1109/SIU.2018.8404179

[31] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. 2017. Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs. *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), 2107–2115.

[32] Lingjing Wang, Jianchun Chen, Xiang Li, and Yi Fang. 2019. Non-Rigid Point Set Registration Networks. *CoRR* abs/1904.01428 (2019). arXiv:1904.01428 http://arxiv.org/abs/1904.01428

[33] David H. Wolpert. 1992. Stacked generalization. *Neural Networks* 5, 2 (Jan. 1992), 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1

[34] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Joshua B. Tenenbaum. 2016. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. In *NIPS*.

[35] Jingjing Xie, Bing Xu, and Zhang Chuang. 2013. Horizontal and Vertical Ensemble with Deep Representation for Classification. *arXiv:1306.2759 [cs, stat]* (June 2013). http://arxiv.org/abs/1306.2759 arXiv: 1306.2759.

[36] Chenhao Xu. [n.d.]. EPDNet: Encoder-less 3D Shape Descriptor Extraction, NYUAD Capstone Seminar. unpublished.

[37] J. Zhang and S. Singh. 2015. Visual-lidar odometry and mapping: low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2174–2181.

[38] Zhirong Wu, S. Song, A. Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and J. Xiao. 2015. 3D ShapeNets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1912–1920.