Supplementary Material: Neural Implicit Embedding for Point Cloud Analysis

1. Robustness of Representation

1.1. Rotational Invariance

We evaluate the robustness of the proposed canonical projection. We compared our projection approach with another commonly used approach to align shapes, which is by conducing principal component analysis (PCA) on surface points themselves. We used the Stanford bunny,which consists of approximately 30000 vertices, and randomly selected $10000, 5000$, and 1000 points from the point cloud 10 different times.

For PCA based alignment, we conducted PCA on each of the sampled point sets directly, and computed the eigenvector corresponding to the largest eigenvalue. We then measured the cosine similarity between all the pairs of the first eigenvectors. For our method, we obtained the distance field by measuring the distance from random sampling points in a unit sphere to the selected object points. We conducted SVD on the matrix created by the sampling point coordinates and its distance function output. We measured the cosine similarity of the first column of $\bar{\mathbf{V}}$, which is the projection to the first axis of the canonical space.

Table 1 shows the mean and the standard deviation of all the cosine similarity values in each method. PCA directly on surface points returns a relatively stable transformation at highly sampled data, but quickly deteriorates when only small subset of the points are small. Our method, in contrast, is robust to both the surface point sampling density and the spatial sampling density, as both the mean and the standard deviation values barely change. This suggests that our alignment method, regardless of the density of the point cloud of the target shape, produces a robust representation. This is due to the fact that points are of the shapes are implicitly embedded into before being transformed into canonical representations. The implicit representation is a non-parametric representation of the shape, therefore, it is unaffected by the resolution of the original point cloud.

methods	10K pts	5K pts	1K pts
PCA mean	0.9997	0.9384	0.7549
PCA st. d.	0.00027	0.1107	0.22898
ours 50K μ	0.9999	0.9999	0.9970
ours 50K σ	4.43E-07	9.73E-07	0.0057
ours 10K μ	0.9999	0.9999	0.9997
ours 10K σ	2.76E-06	4.66E-06	0.0003
ours 5K μ	0.9999	0.9999	0.9995
ours 5K σ	1.05E-06	1.94E-05	0.0005

Table 1: Comparison of the deviation of principle axes. We compared the mean pairwise cosine similarity and the standard deviation of the vector transforming the input to the first principle axis. Applying PCA on point cloud data is robust when the points are dense, but quickly deteriorates. Our method, on the other hand, uses a spatial representation, making the decomposition robust to point density.

sampling density	256	512	1024	2048
k=256	0.0435	0.0485	0.0545	0.0584

Table 2: RMSE between distance values reconstructed using ELMs and the ground truth distances.

1.2. Accuracy of ELM Embedding

We evaluate how accurately ELM captures the distance field. We train an ELM to capture the distances from random sampling points scattered within a sphere to the closest points in the point cloud. We then compare the distance values that the trained ELM returns against the ground truth values. We used the Stanford Bunny as the target point cloud in this demonstration. Dimension of the ELM weights are set at 256, and the samplings points and the point cloud is scaled to a unit sphere.

As can be seen from Table 2, the margin of error of reconstruction is relatively small, considering the fact that the ELM has only one hidden layer, and that the weights $\mathbf{W}$ are fixed. We therefore claim that using ELM is adequate to be used to encapsulate local portion of the distance fields.

2. Detailed Results from Classification

2.1. Effect of Elements

We provide detailed results from the classification experiment in the main manuscript. We varied the elements that affects the outcome of the results. They are the radius of

	number of sample points											
	256			512			1024			2048		
	dimension of ELM weight											
k	128	256	512	128	256	512	128	256	512	128	256	512
0.1	92.51	92.73	91.63	93.28	93.17	92.51	91.30	93.28	92.84	92.07	92.84	92.51
0.2	94.49	94.93	94.60	95.37	95.37	94.27	95.04	94.82	95.15	94.27	94.82	94.16
0.3	94.93	94.93	94.60	95.82	95.92	95.26	95.04	95.26	95.26	94.27	95.26	94.82
0.4	94.38	94.60	94.49	94.60	95.26	95.37	94.93	94.49	94.71	94.60	95.04	95.37

Table 3: Accuracy for ModelNet 10 with different elements.

	number of sample points											
	256			512			1024			2048		
	dimension of ELM weight											
k	128	256	512	128	256	512	128	256	512	128	256	512
0.1	91.73	91.32	91.12	91.25	91.37	91.32	91.37	91.82	91.29	90.92	91.90	91.29
0.2	91.45	90.92	90.84	90.96	91.98	91.77	90.76	91.61	91.00	91.00	91.21	91.12
0.3	90.84	91.16	91.25	91.86	92.67	92.06	90.72	91.77	91.29	91.17	91.09	91.41
0.4	90.96	91.61	91.21	91.18	91.82	91.61	91.09	91.25	91.09	90.69	90.64	91.05

Table 4: Accuracy for ModelNet 40 with different elements.

the sampling spheres, number of samplings points within a sampling sphere, and dimension of the ELM bases. Note that points and normals are concatenated with the ELM parameters to obtain the results.

We provide the results obtained from ModelNet 10 and ModelNet 40 in Tables 3 and 4, respectively. Median value of 5 attempts were used to fill the chart. There is no clear trend in the elements, except for the fact that all the elements cannot be too small or too large to achieve optimal accuracy.

As the dimension of the input to the classification network changes depending on the size of the ELM, the network structure may be keeping some of the accuracies in the table low. For the purpose of comparison, we used the same structure for all the data. Adding or subtracting layers for the classification network to suit the representation may improve some of the results in the table.

There is also a high likelihood that higher accuracy can be obtained by preparing different number of sampling points, or dimension of ELM that was not used in this experiment. We will continue to pursue the suitable combination of elements, as there are infinite number of possible combinations. Automatic identification of best possible elements would be considered in future work.

3. Segmentation Results

Finally, we present some of the results from the point cloud segmentation experiment. We use the ShapeNetCore Part dataset for the experiment. We use the same setting from the classification experiment to obtain the results. The number of sampling points, ELM dimension, and sampling sphere radius, are 1024, 256, and 0.3, respectively.

We first show some of the more successful segmentation results. As can be seen from Fig. 1, the successful results provide accurate labels to parts without any confusion. Our method is especially successful at identifying labels of structures that sticks out, for example, the wings of airplanes, the legs of chairs, and so on. This is due to the fact that our embedded feature encapsulates the local region around each point. Therefore, the context around each point is included in the input, which facilitates the assignment of part labels to these prominent structures on objects.

We then introduce some of the less successful results. Fig. 2, shows the objects with less successful segmentation predictions. As can be observed from the images, Despite being able to assign different labels to separable parts, the assigned labels in these cases are wrong. This is due to the fact that the method recognized the parts, but observed the entire object as something different, leading to parts from that class. This is caused by concatenating the global signature from the model to each points on the model as the input feature to segmentation classifier. Avoiding such concatenation may avoid these issues, and would be one of the future directions for the task of assigning pointwise labels.

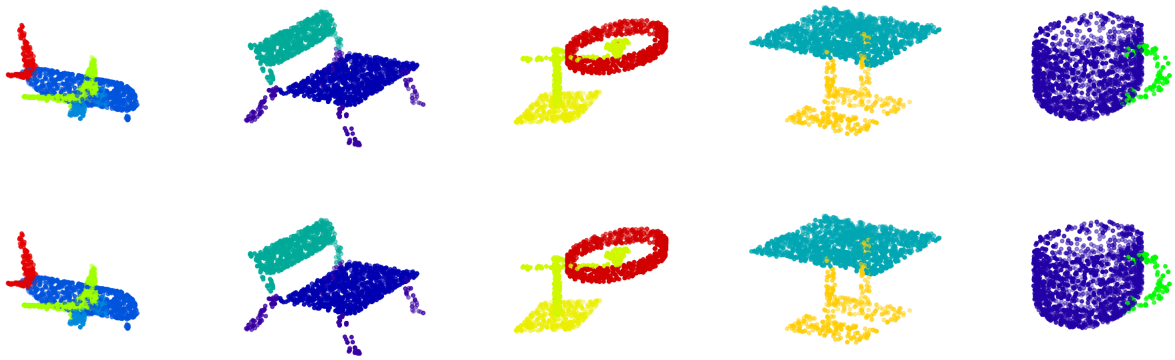

Figure 1: Accurate segmentation results. Top row: Predicted labels from the proposed method. Bottom row: Ground truth.

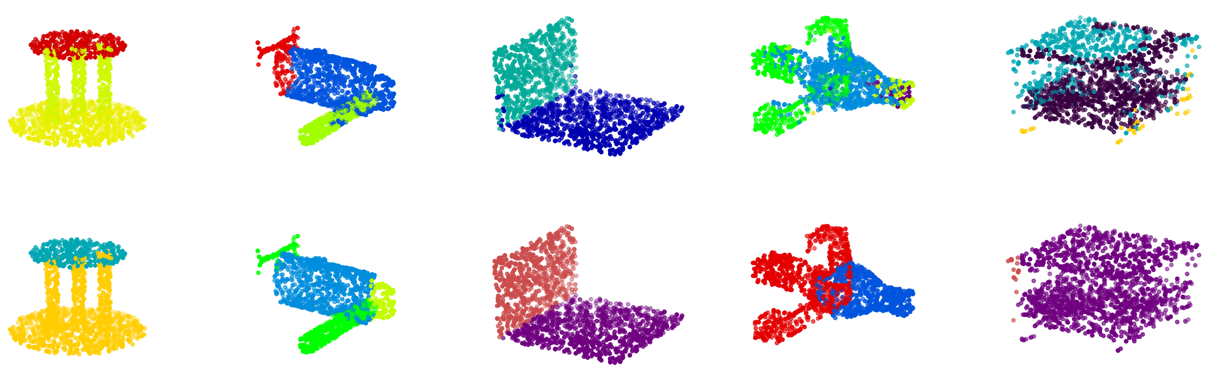

Figure 2: Erroneous segmentation results. Top row: Predicted labels from the proposed method. Bottom row: Ground truth.