

Figure 4: Illustration of the constructed autoencoder. Green blocks represent a normal bottleneck ResNet block, orange ones are blocks with a decrease factor of $f_d = 6$, and blue blocks are blocks with a transposed convolution and $f_u = 6$. The numbers above and below indicate the size of the input tensor for the corresponding block.

A Autoencoder Architecture

In the network structure of the encoder, the compute path of those ResNet blocks at first decreases the number of channels per node to make the main convolution less computationally heavy (bottleneck ResNet blocks). This happens via a convolution processing only the individual nodes. For the second convolution, behind every point the vectors to its k -nearest-neighbors are gathered. Then, all point vectors are processed by a convolution with kernel sizes of (1×1) . In the case of a stride greater than one, only the selected points are processed. The last convolution inflates the channel dimension again to the desired output size. The identity path of a ResNet block calculates the mean of the node features in the neighborhood of the node if stride is employed. Alternatively, if no stride is employed, it simply passes the feature map.

Two such blocks with a stride of one followed by a block with a higher stride constitute a level with the same number of nodes being considered in the computations. Pang et al. [13] choose a code word of length 512 for point clouds of size 2,048. To obtain a similar sized code word from a two level deep network, a decrease factor of 6 is chosen for both levels, resulting in 57 points with 9 channels each and a code word length of 513. This yields a balanced relation between points represented in the code word and features describing them.

The decoder is conceptually built in the same way. Only in the third block of one level, the last convolution operation is replaced with the transposed convolution operation. Further, the identity path samples new points by taking the mean vector between p_i and one of its neighbors p_j . The complete architecture can be seen in Figure 4.

B Detailed Ablation Results

In table 3 we show the above mentioned results of the conducted ablation studies. As stated before the resulting chamfer distances only vary slightly among the tested configurations.

In figure 5 we additionally show two randomly selected point clouds from the ModelNet40 [23] dataset. In the left column we scattered the first three features of the code word representation of the point cloud that is shown in the right column. We find that those three features, if scattered in three dimensional space like this, resemble the rough shape of the input point cloud. We assume this can be attributed to the ResNet architecture included in our autoencoder. The direct put through of the input data seems to cause the model to keep the rough three dimensional shape in its first three code word features. This information despite being useful for the reconstruction of the point cloud may not be insightful for a linear SVM hinting at a reason for the improved performance in the case of an increased cc. At the same time having the rough shape of the input point cloud represented inside the code word seems to enable better point cloud reconstruction compared to competitive models.

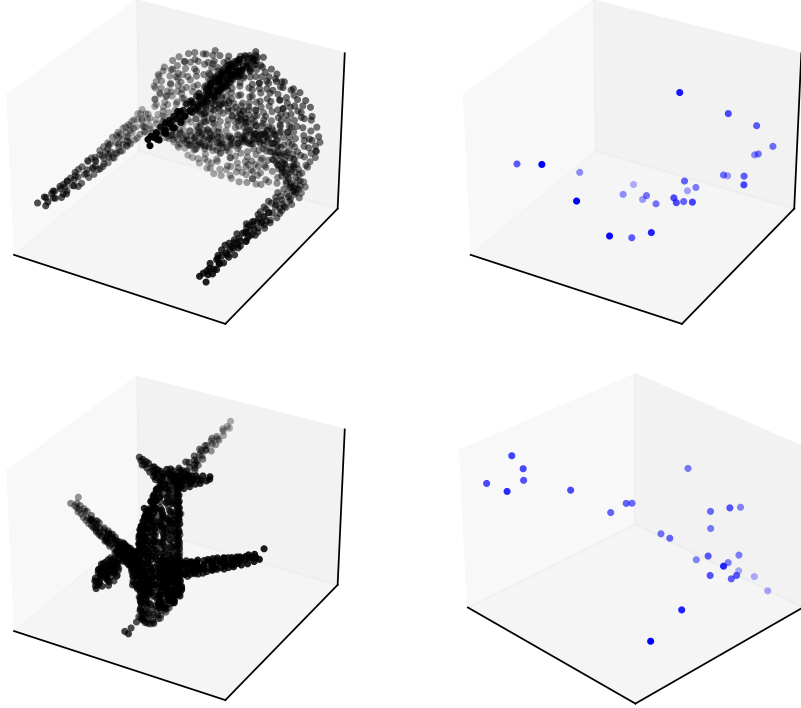


Figure 5: Two selected 3d objects from the ModelNet40 [23] dataset. In the left column one can see the original object point cloud in black. In the right column we see the point cloud constructed from the first three code word channels of the autoencoder which was configured according to the first row of table 3.

Table 3: The performance values of the different models on the ShapeNet [4] data set given in terms of the extended chamfer distance (Equation 5) multiplied by factor 100 for better readability. In the same setting FoldingNet achieves a score of 2.09.

$f_{d,1}$	$f_{d,2}$	$f_{u,1}$	$f_{u,2}$	k	code word channels (cc)	FPS	chamfer distance
6	6	6	6	10	9	—	1.81
8	4	4	8	10	8	—	1.83
4	8	8	4	10	8	—	1.80
6	6	6	6	8	9	—	1.84
6	6	6	6	16	9	—	1.82
6	6	6	6	10	9	✓	1.81