

3D Conv Nets for Robotic Manipulation

Chad DeChant, Joaquín Ruales, Jake Varley

Outline

- 3D Spatial CNNs
 - Extended 2D to 3D
- Grasping
 - Heatmap Approach
 - Shape Completion Approach
 - Comparison

2D Conv Filters



*

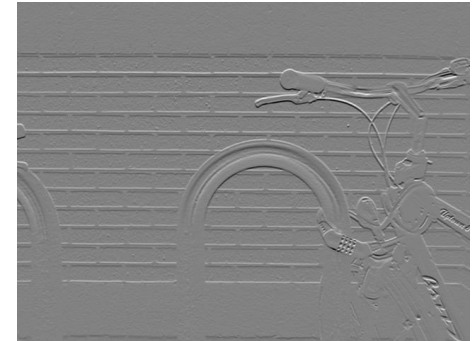
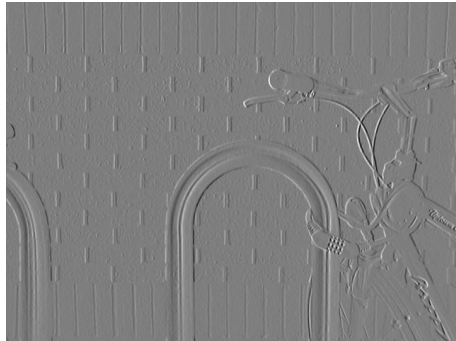
*

-1	0	+1
-2	0	+2
-1	0	+1

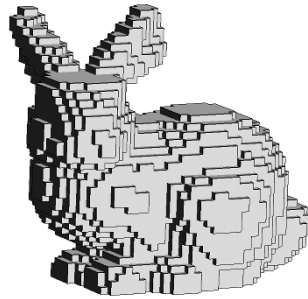
+1	+2	+1
0	0	0
-1	-2	-1

x filter

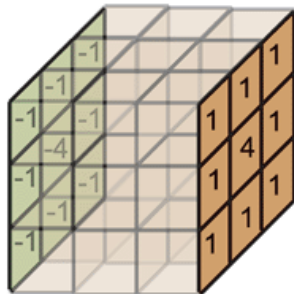
y filter



3D Conv Filters

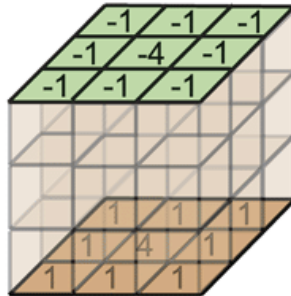


*



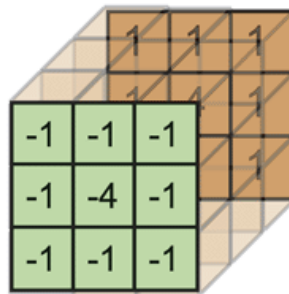
x filter

*



y filter

*



z filter

3D Filters in Theano

- We Implemented the following for the Keras Theano library
 - 3DConvolution
 - Theano conv3d2d (BZCXY format)
 - 3DMax pooling
 - Created max_pool_3d using max_pool_2d

2D Conv



**3x3
Image**

1	2	3
4	5	6
7	8	9

**2x2
Filter**





=



Sum of 1D Convs



**2x2
Output**

(1 | 2 | 3 *  )

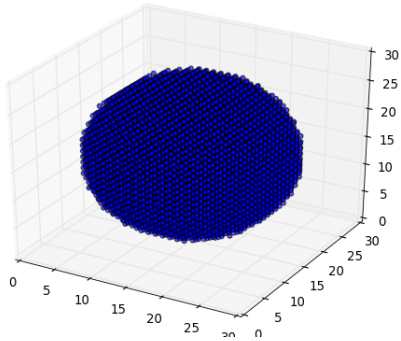
+ (4 | 5 | 6 *  )

(4 | 5 | 6 *  )

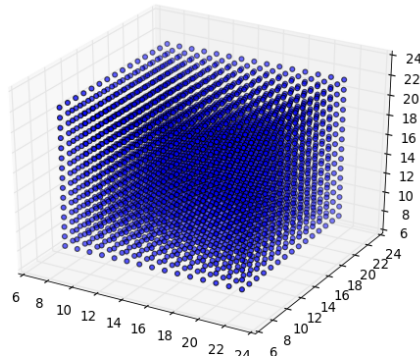
+ (7 | 8 | 9 *  )

3D Spatial CNN Test Data

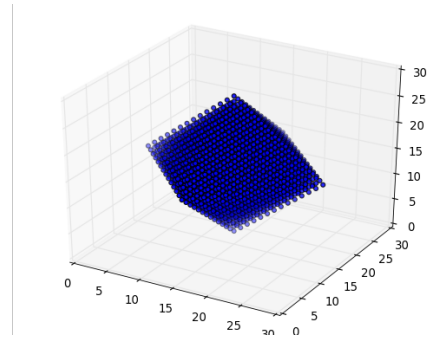
- Random size geometric shapes



sphere



axis-aligned cube



rotated cube

Cost Function

Sphere

Fully connected logistic layer

Max Pooling

Conv layers (16 3x3x3, relu)

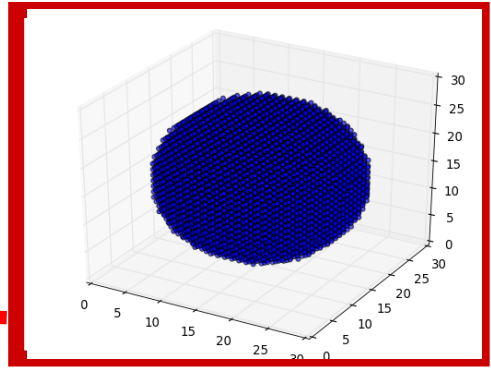
Max Pooling

Conv layers (8 5x5x5, relu)

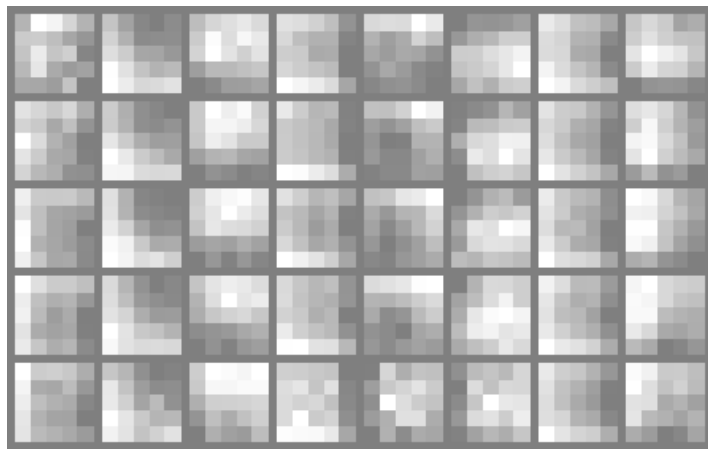
INPUT (28x28x28)

Sphere

Target output



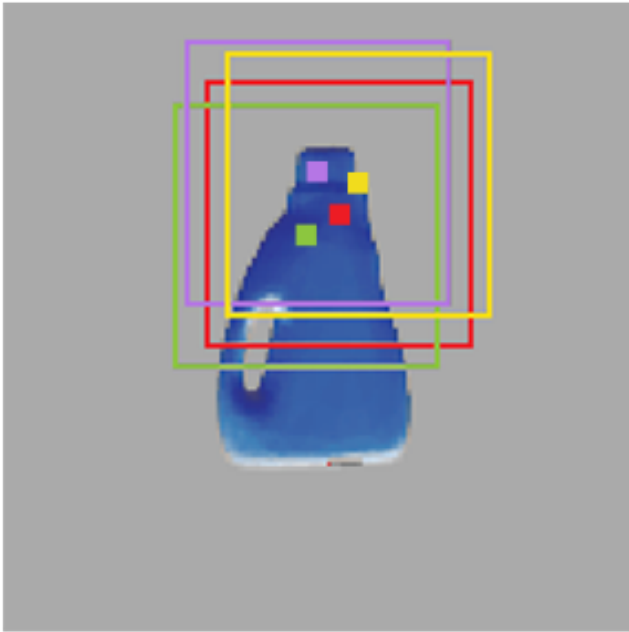
Trained Layer 0 Filters



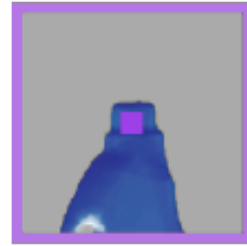
1 2 3 4 5 6 7 8

8 5x5x5 filters

Grasping Heatmaps



Palm



F0

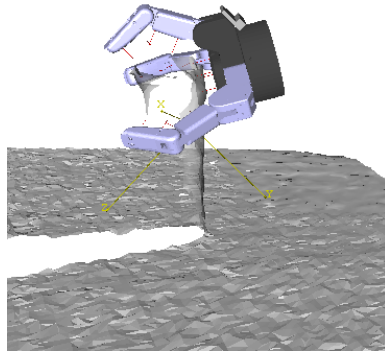


F1

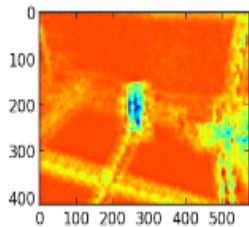


F2

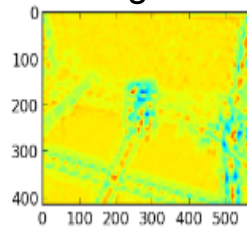
Grasping Heatmaps



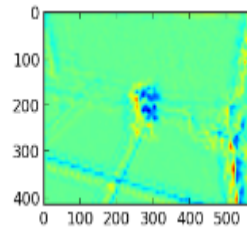
Palm



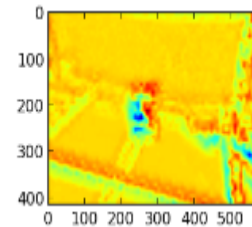
Finger 1



Finger 2



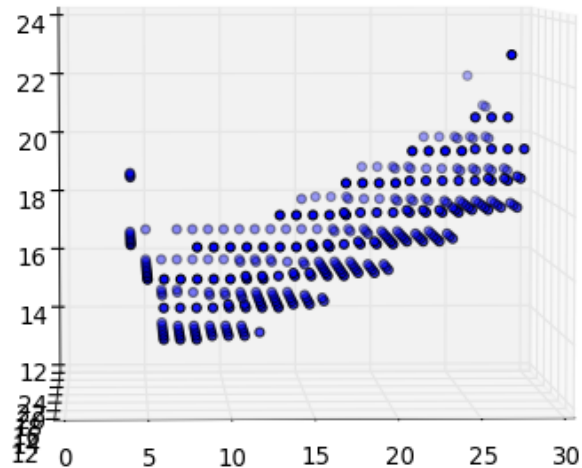
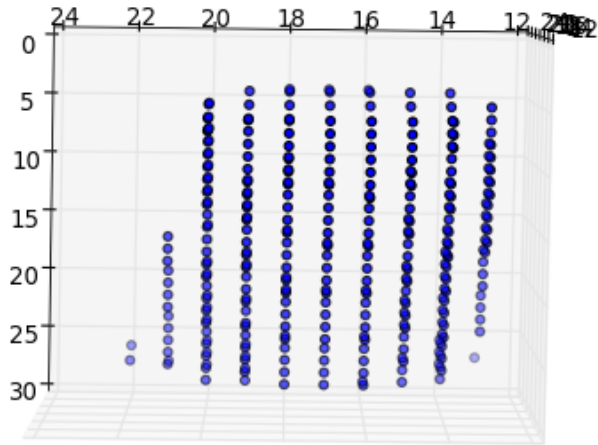
Finger 3



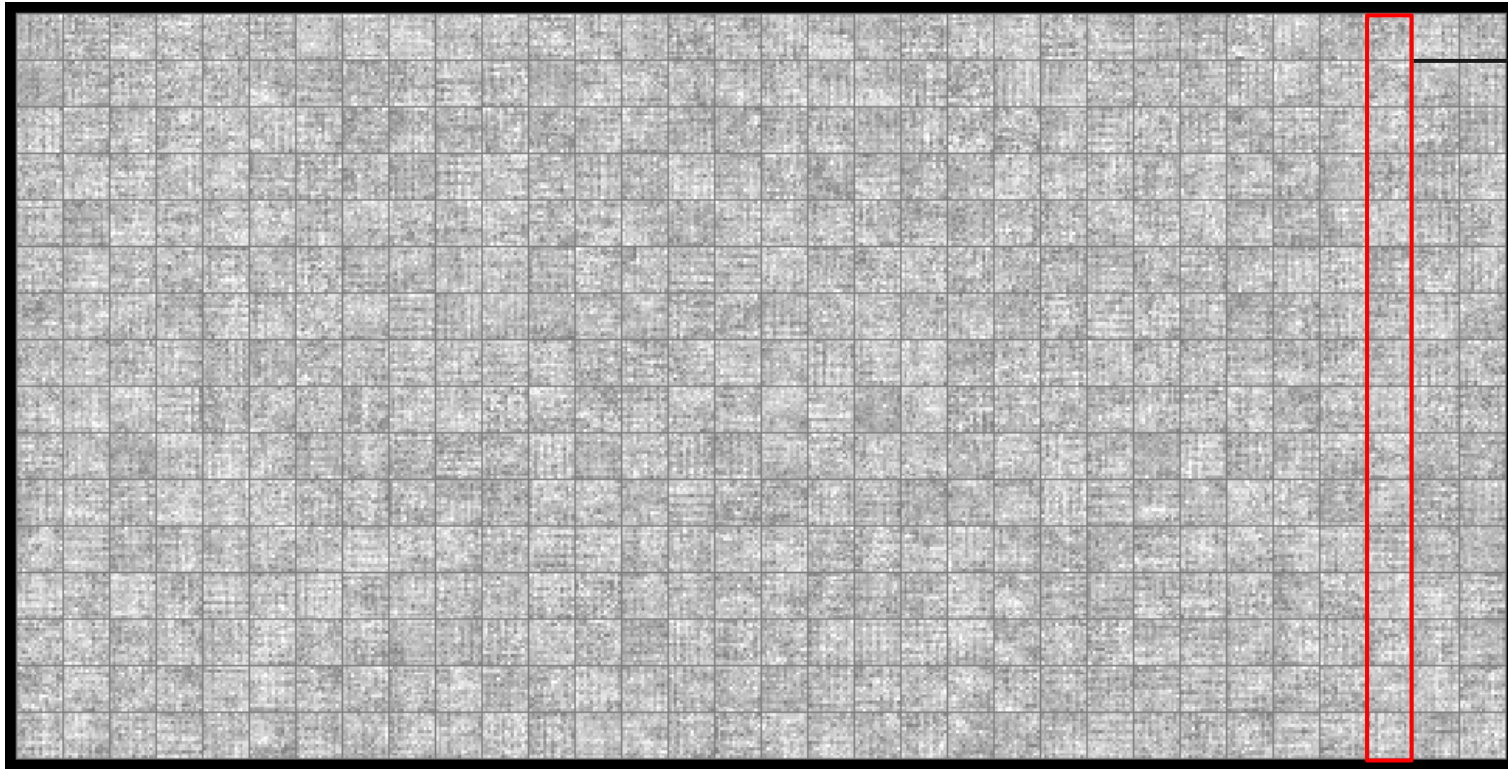
Grasping Heatmaps

Patch: 32x32x32

Label: 1x1x1x nheatmaps



Trained Layer 0 Filters

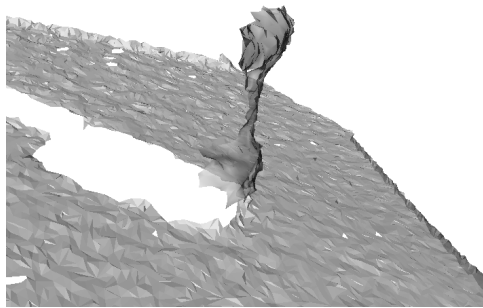
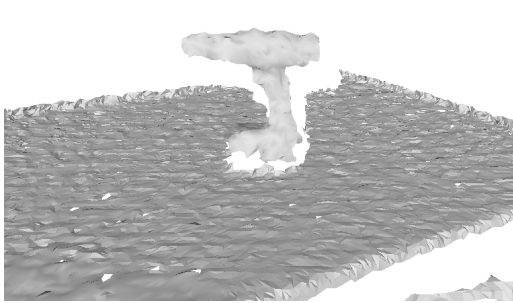


1 16x16x16
filter

Another Approach...

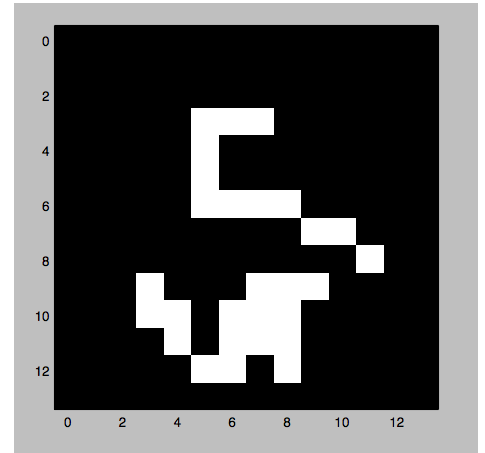
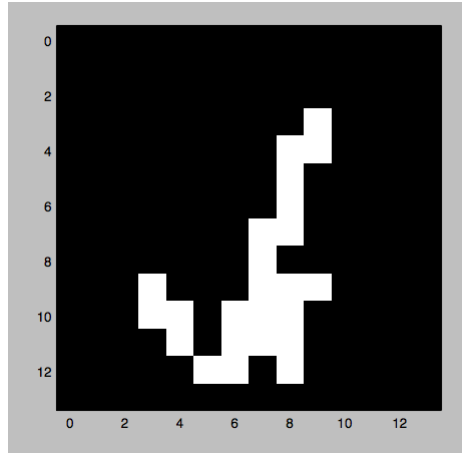
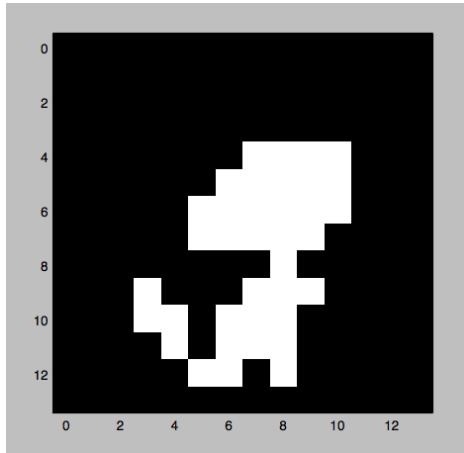
Shape Completion

Complete the shape, then plan grasps on it

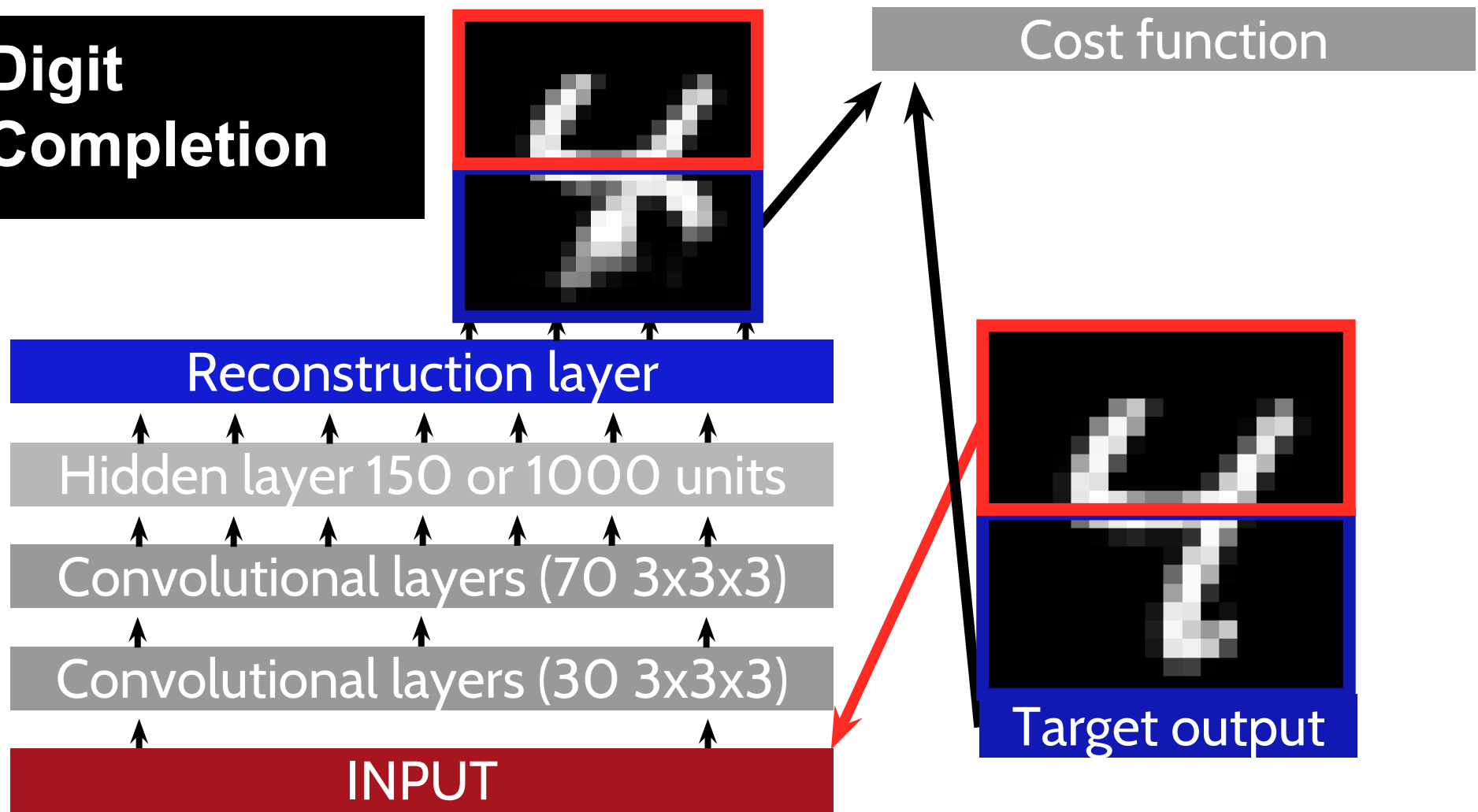


Shape Completion

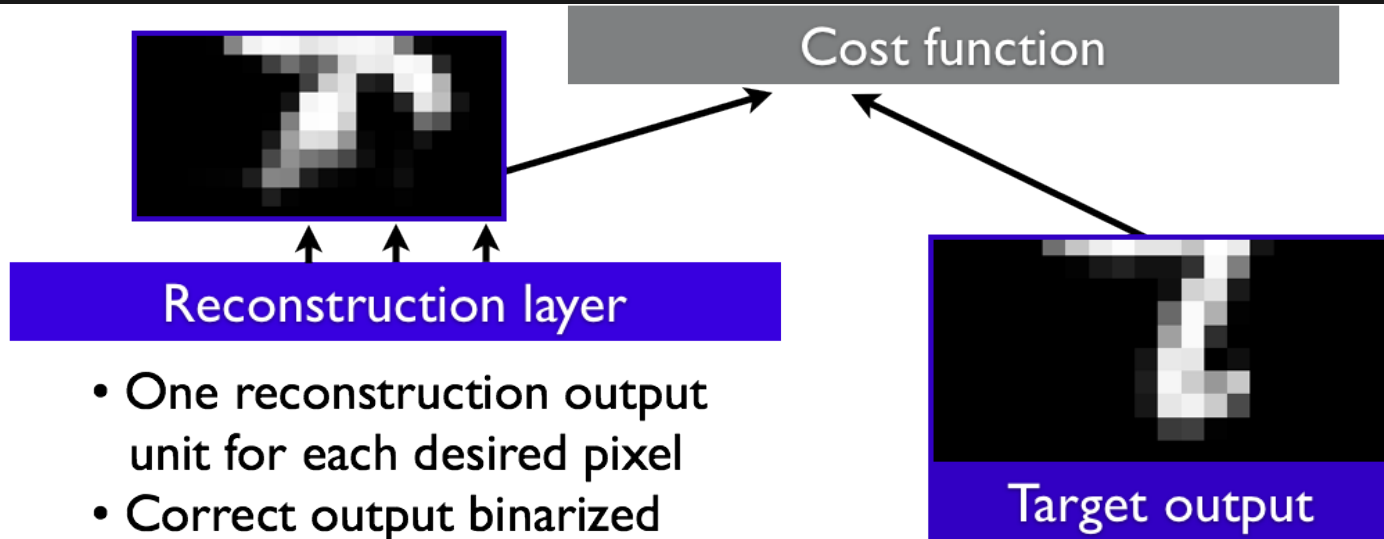
- Is shape completion feasible?
- Use MNIST



Digit Completion



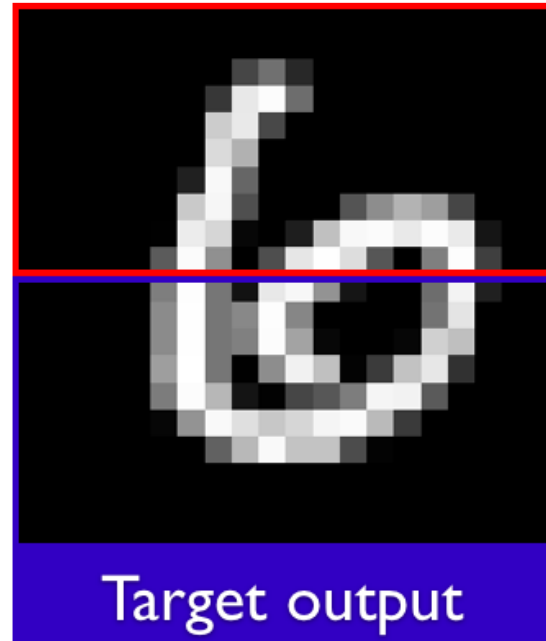
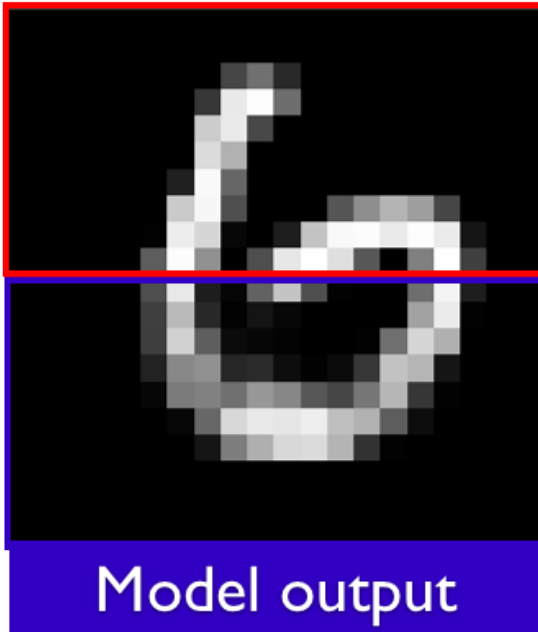
Digit Completion



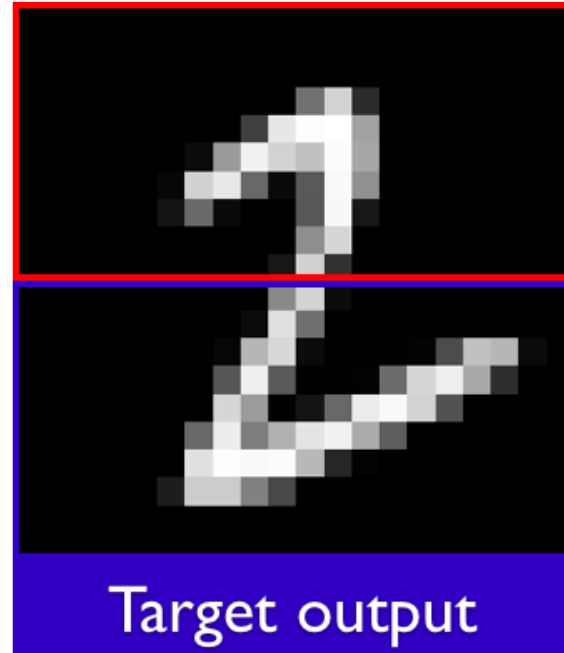
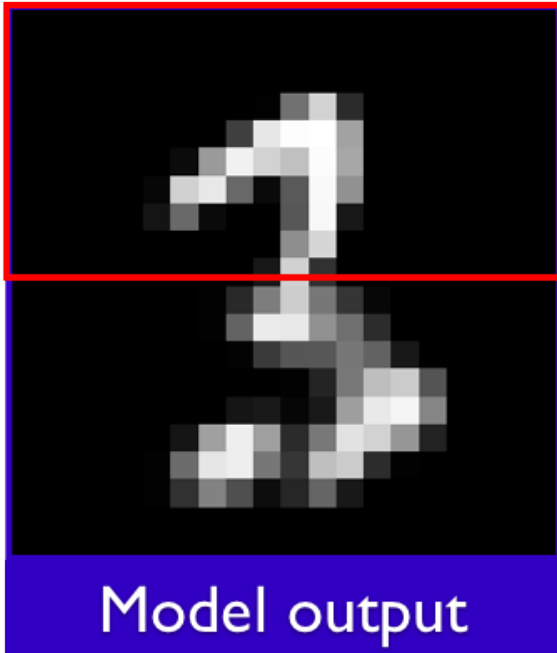
- One reconstruction output unit for each desired pixel
- Correct output binarized
- Sigmoid units encouraged to be close to either 0 or 1

```
def cross_entropy_error(self, y):  
    cost = - T.sum(y * T.log(self.output) + (1 - y) * T.log(1 - self.output), axis=1)  
    return T.mean(cost)
```

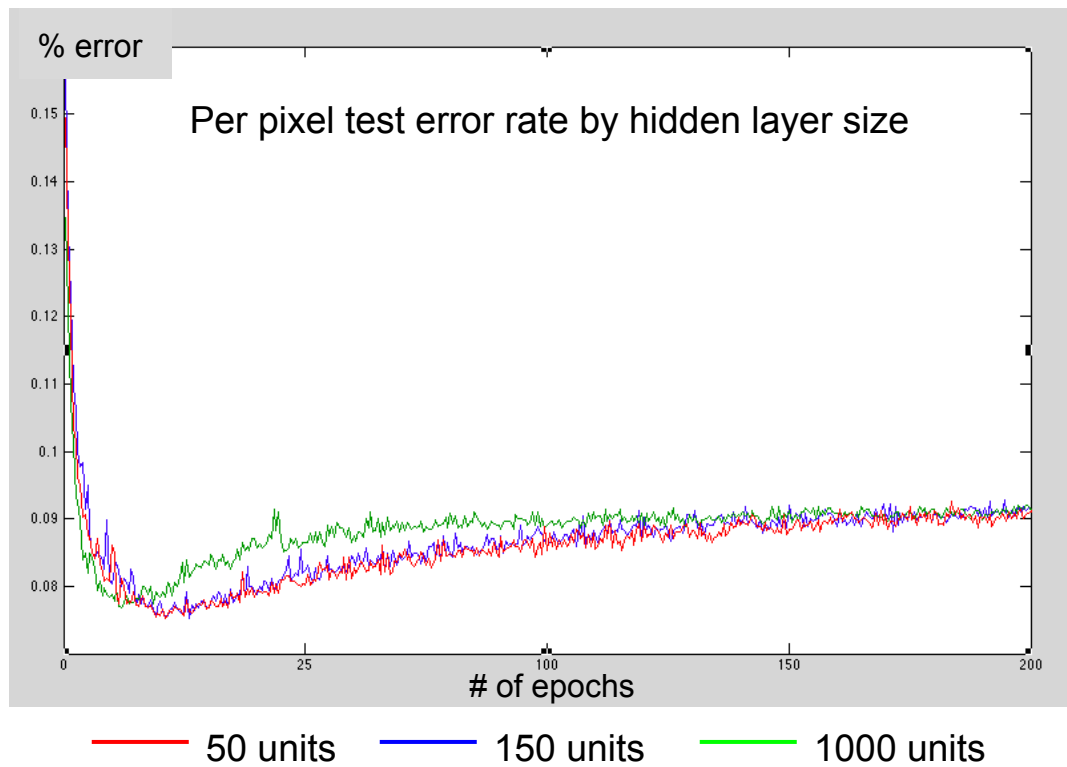
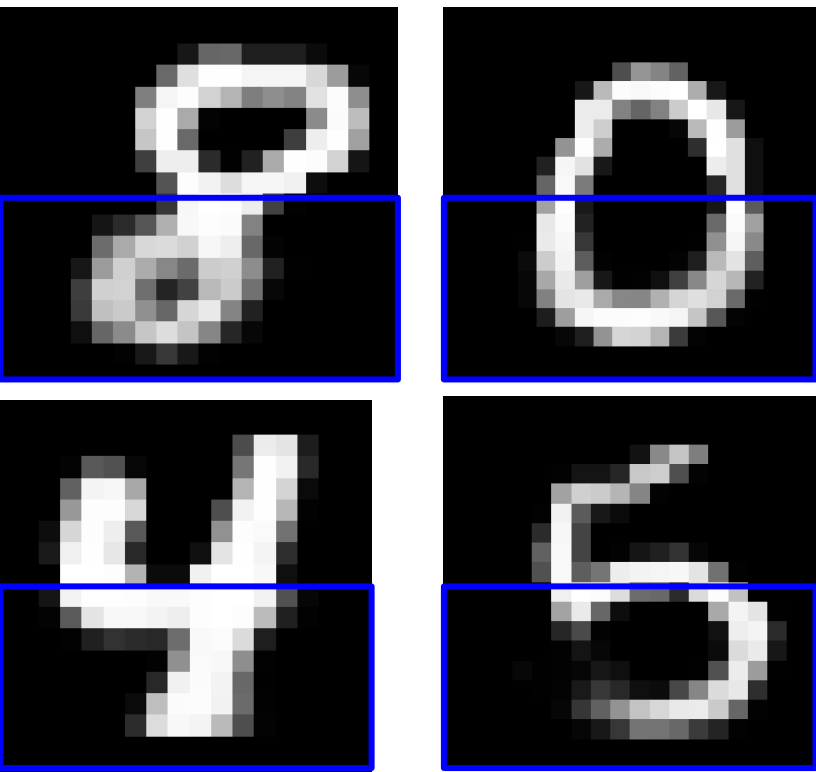
Digit Completion



Digit Completion

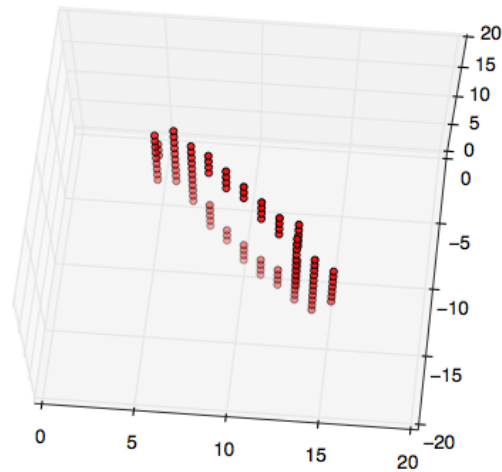
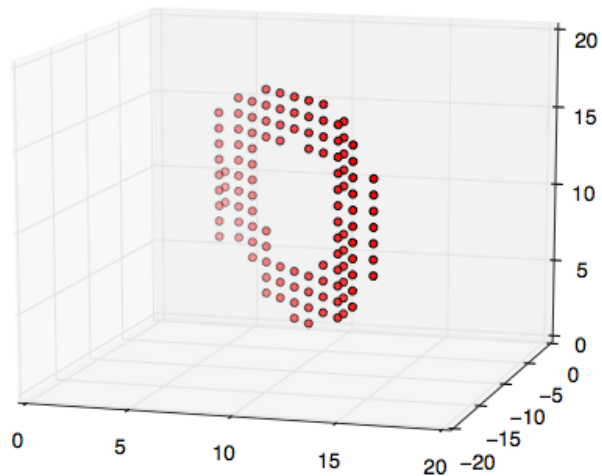
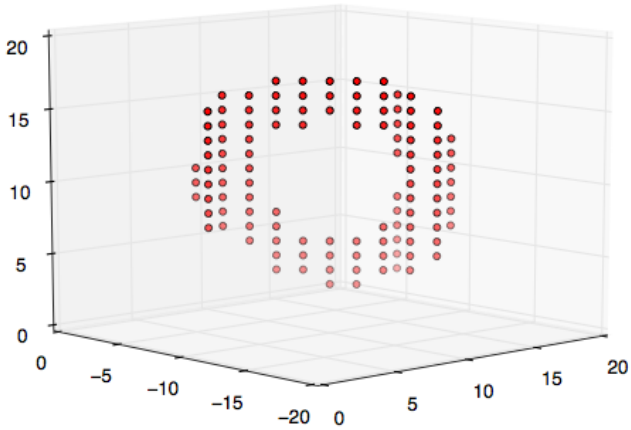


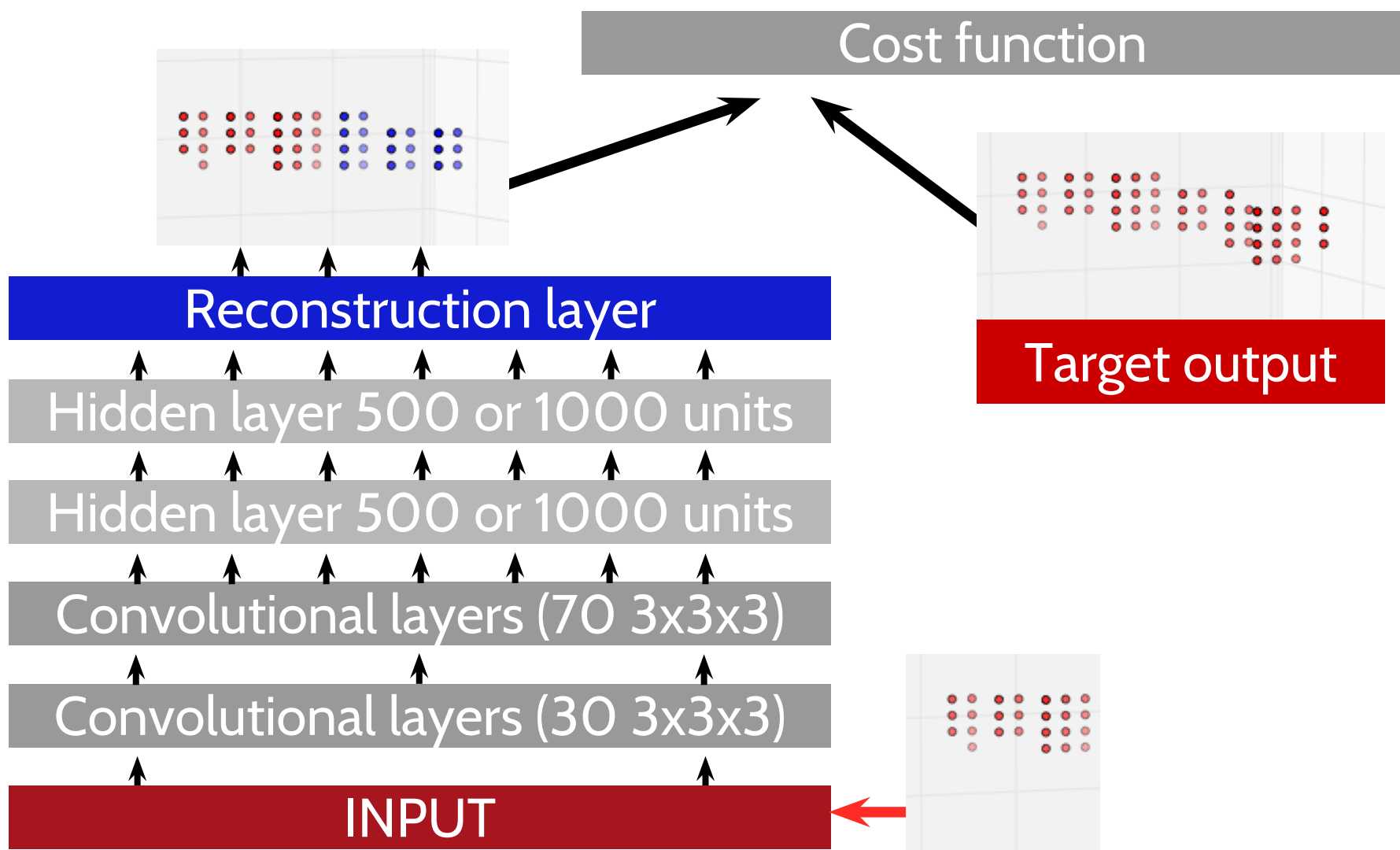
Digit Completion



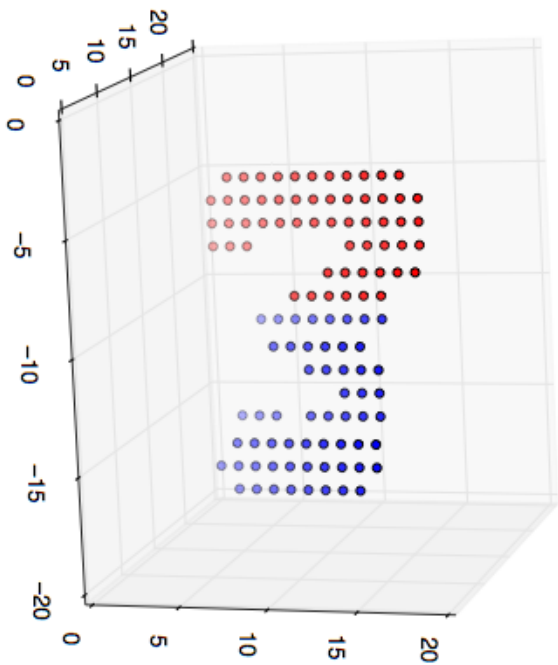
3D digit completion

MNIST in three dimensions

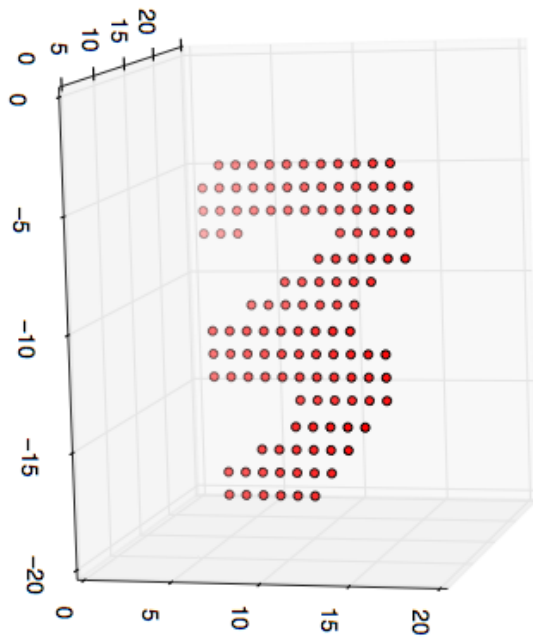




3D Digit Completion

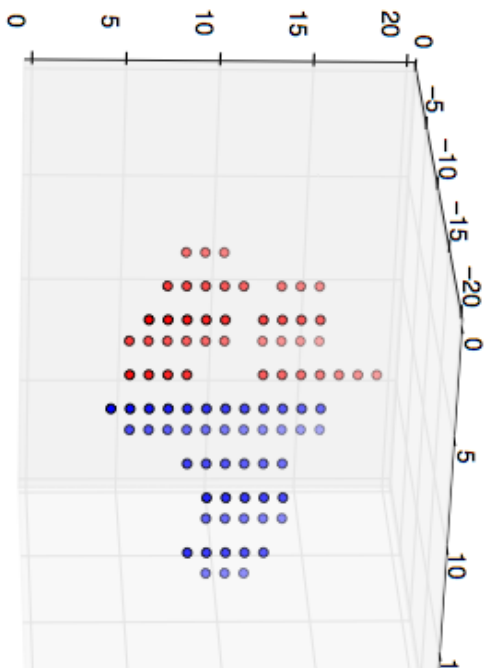


Model Output

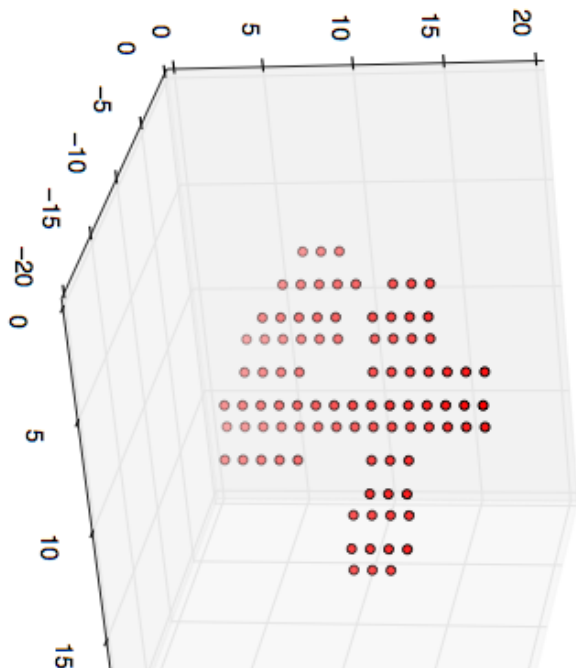


Target Output

3D Digit Completion

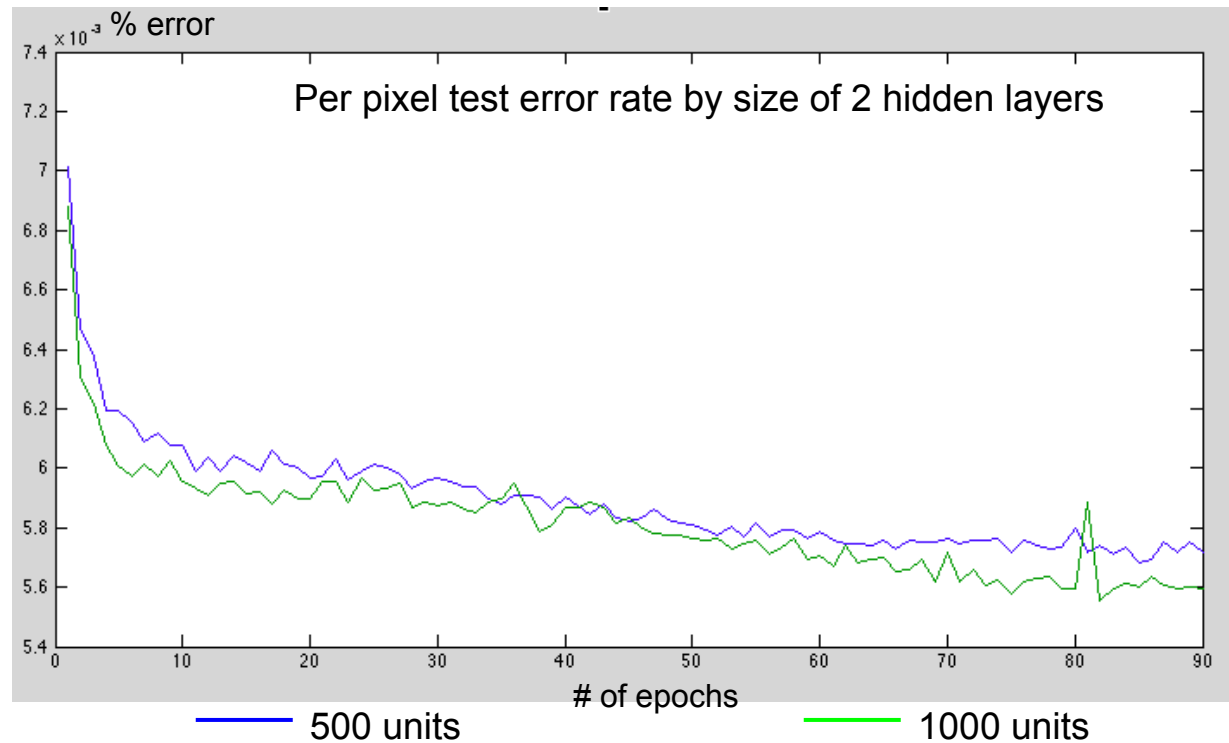
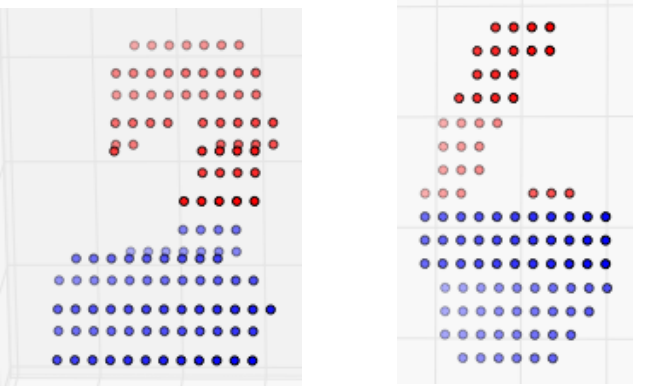
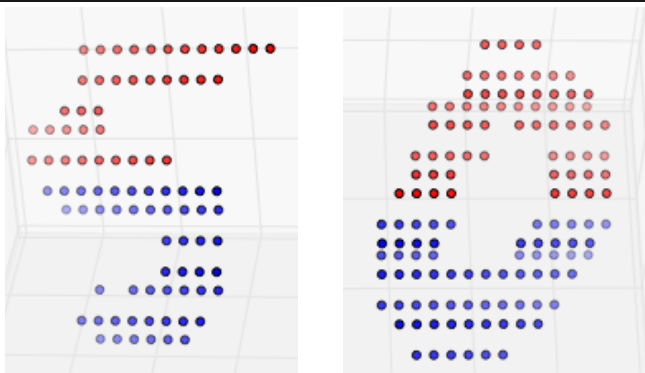


Model Output

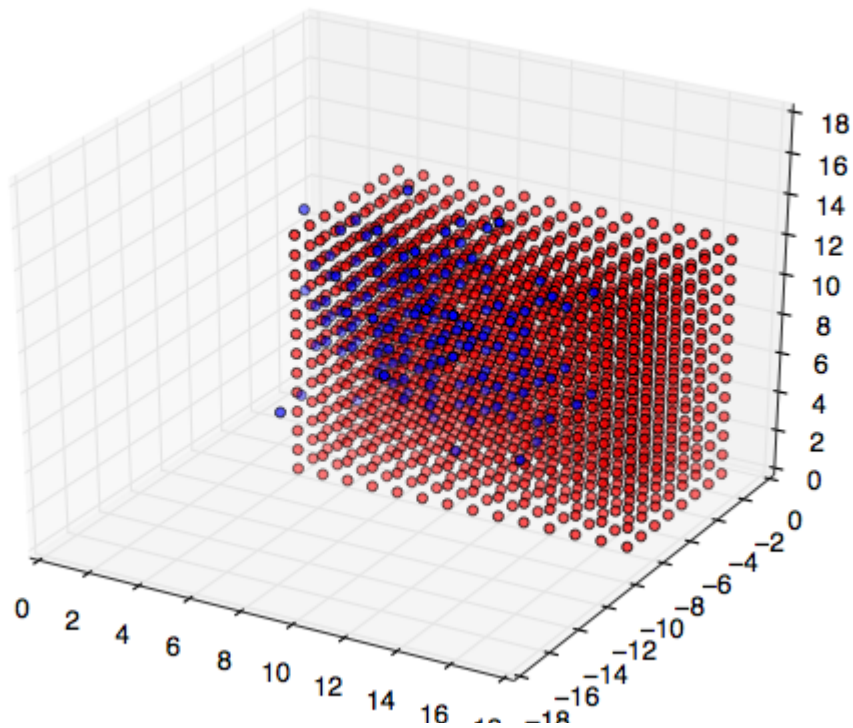


Target Output

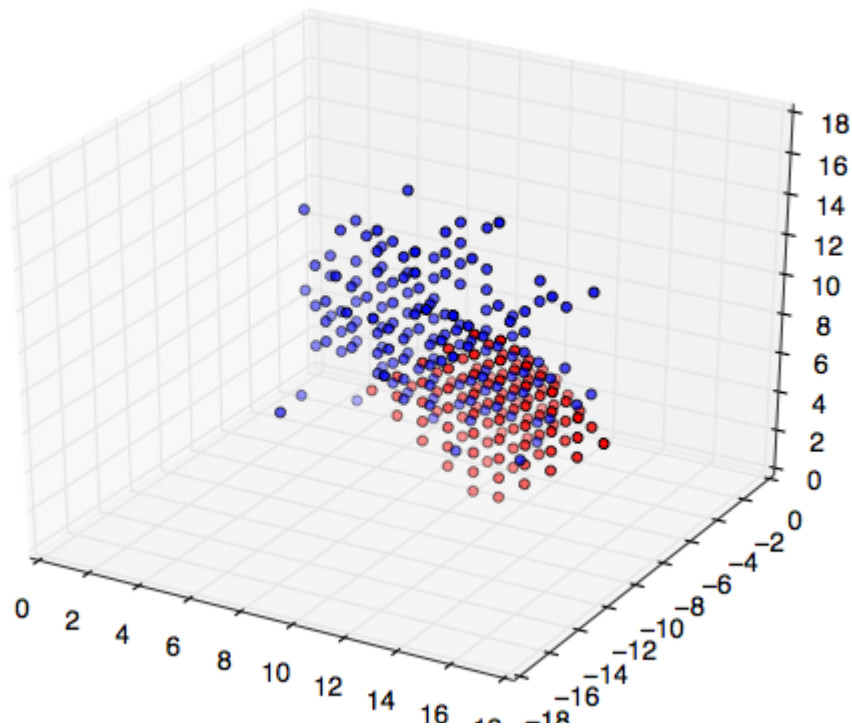
3D Digit Completion



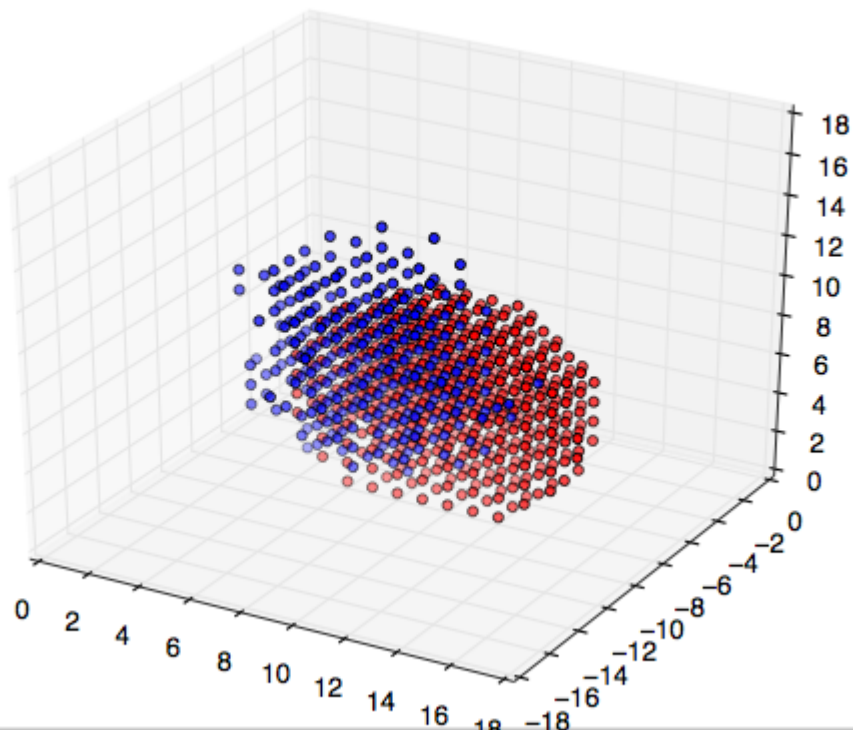
Synthetic shape completion



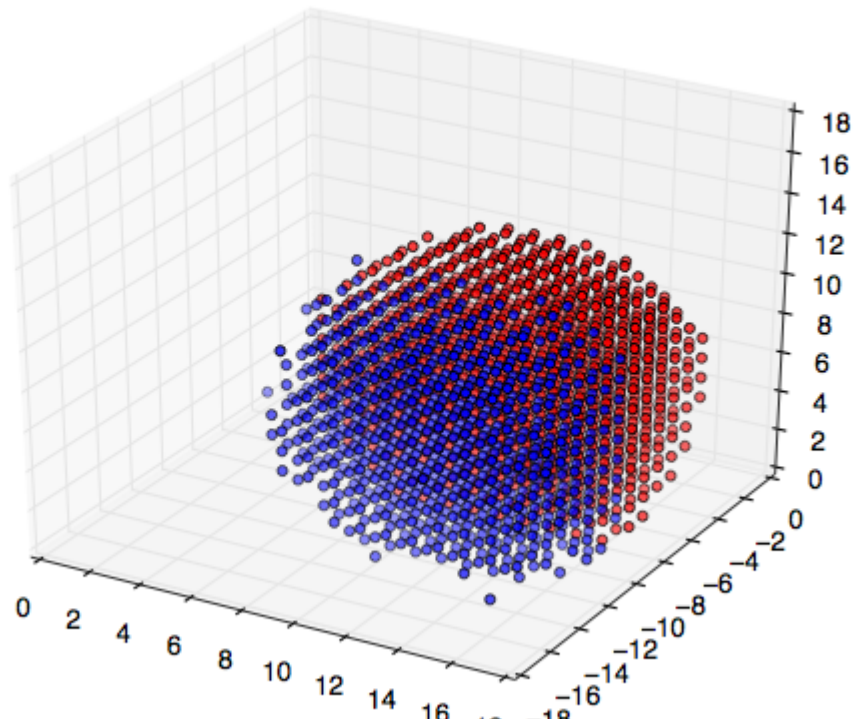
Synthetic shape completion



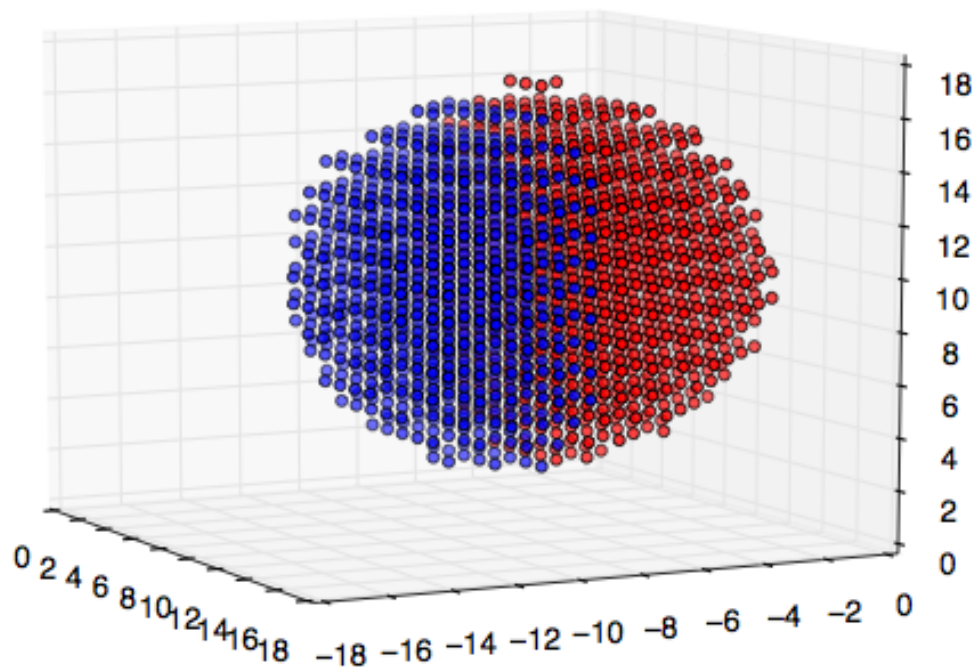
Synthetic shape completion



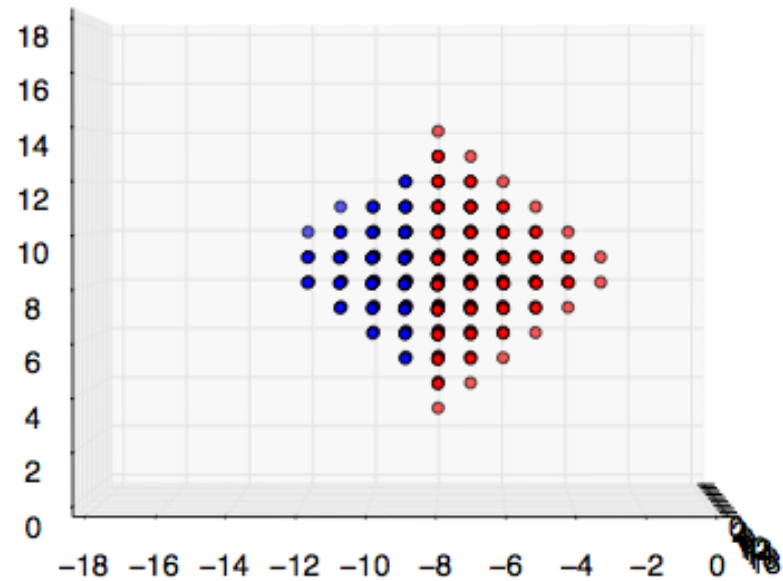
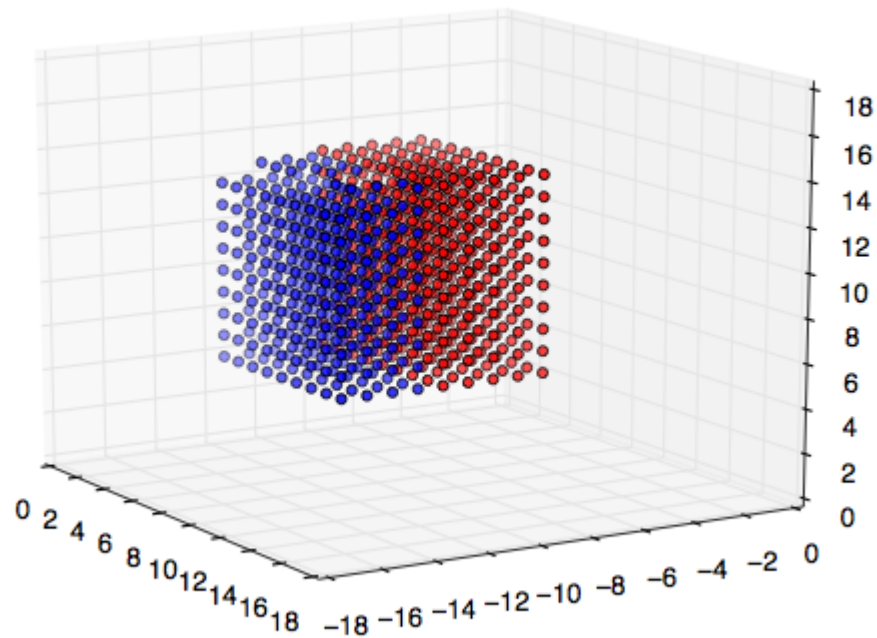
Synthetic shape completion



Synthetic shape completion

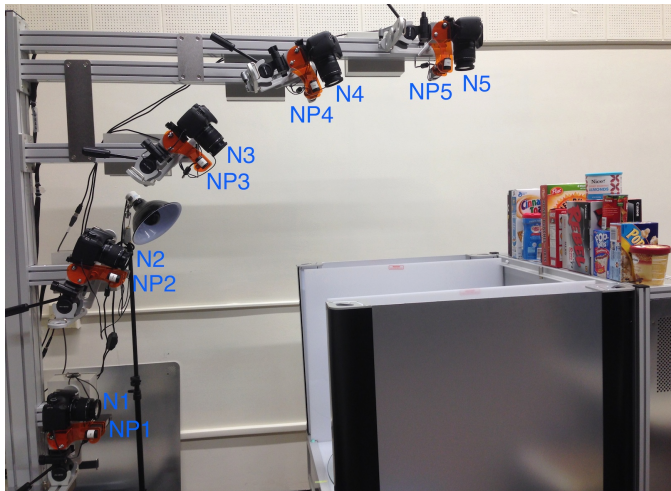


Synthetic shape completion



Training Data

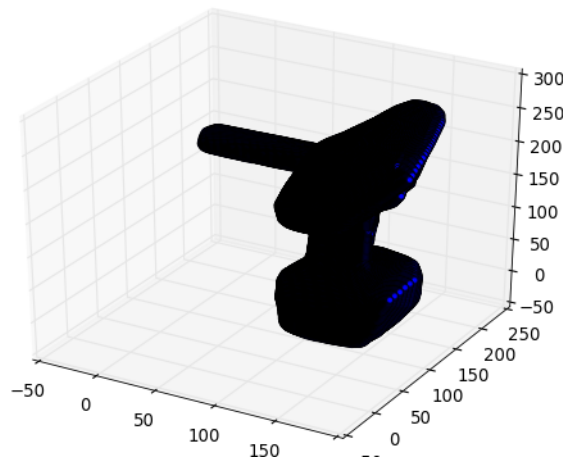
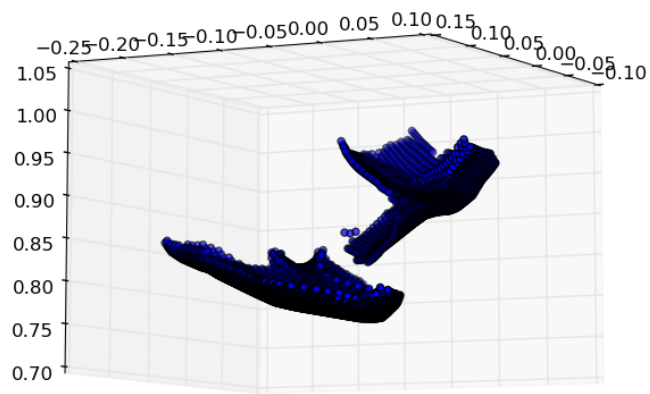
- Household objects, Modelnet, BigBird



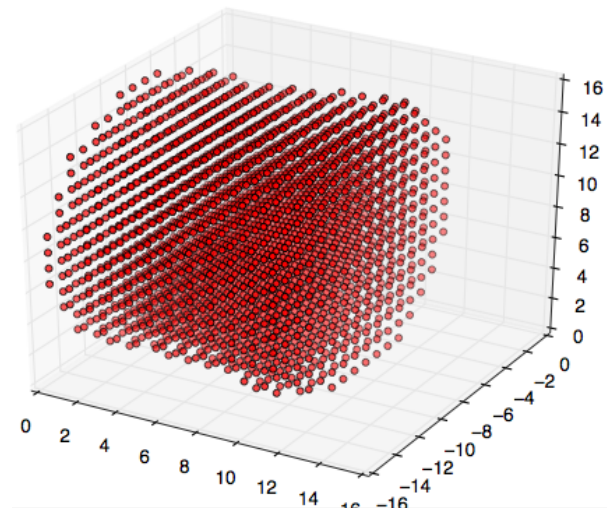
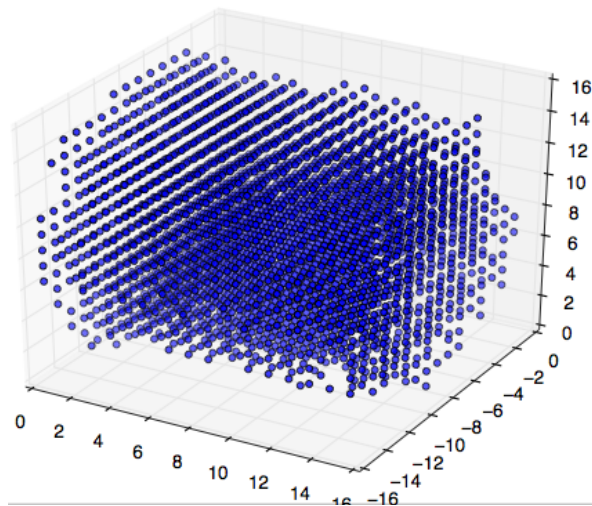
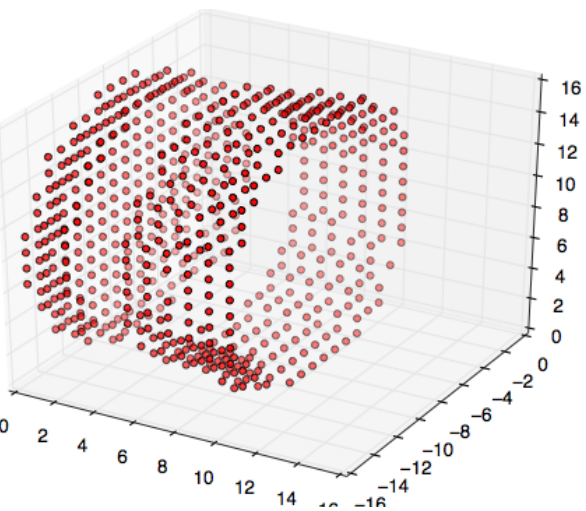
Shape Completion Data



Voxelization



3D Shape completion

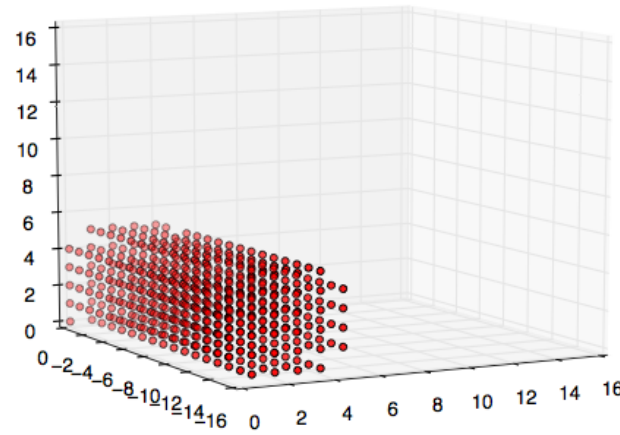
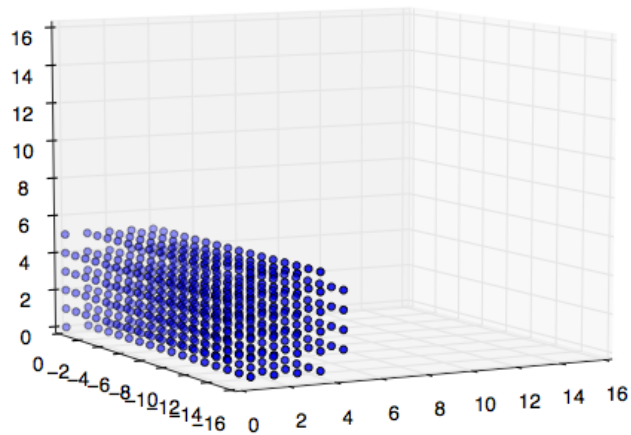
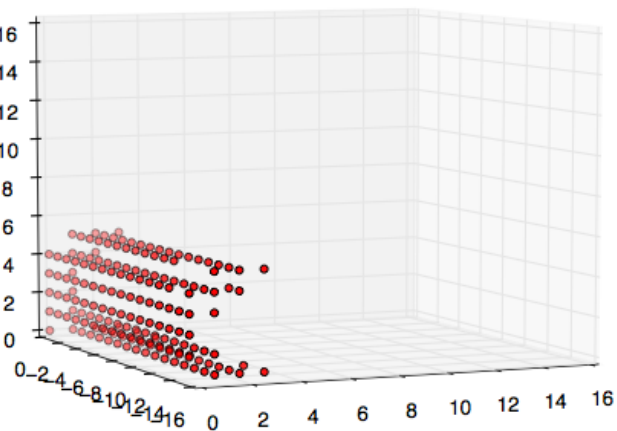


Model Input

Model Output

Target Output

3D Shape completion



Model Input

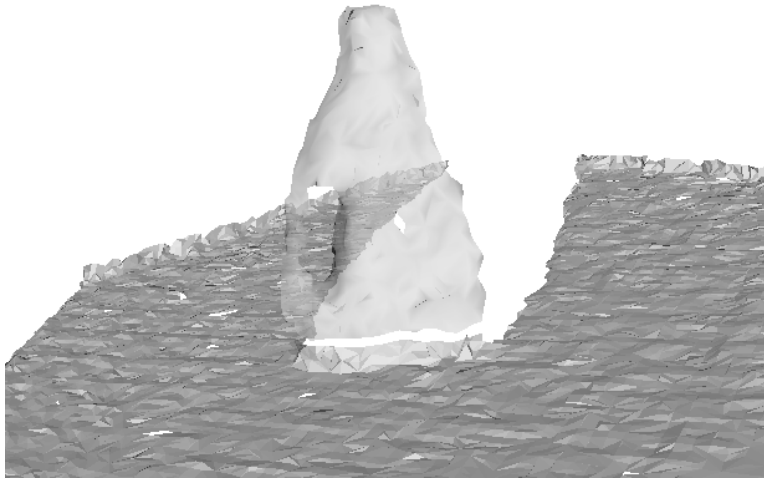
Model Output

Target Output

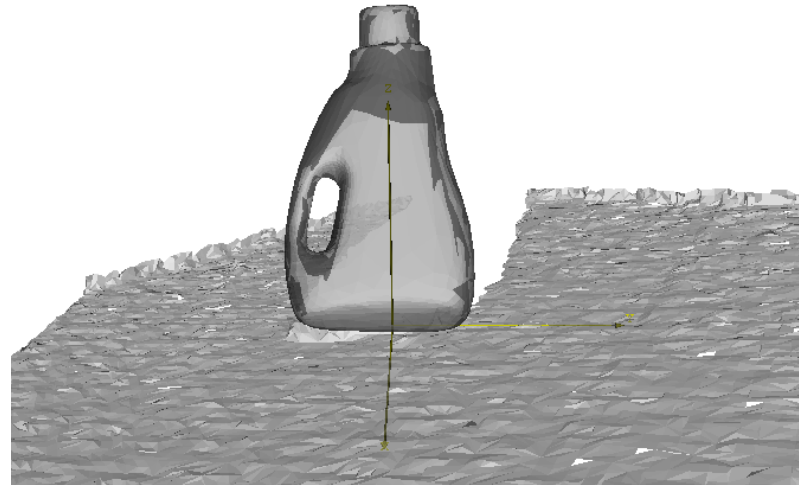
Grasp Comparisons

Heatmap vs Shape Completion Grasps

1) generate grasps



2) compare ground truth grasp qualities



Conclusion

- 3D Spatial CNNs
 - Extended 2d to 3d
 - Keras Library
- Grasping
 - Heatmap Approach
 - Shape Completion Approach
 - Comparison

Thank you

