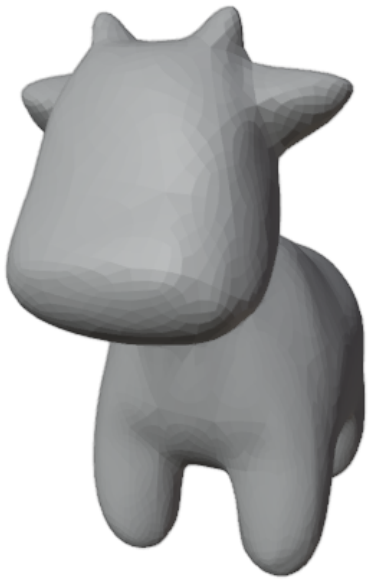
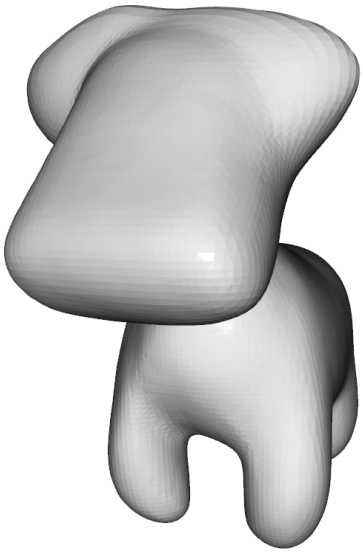
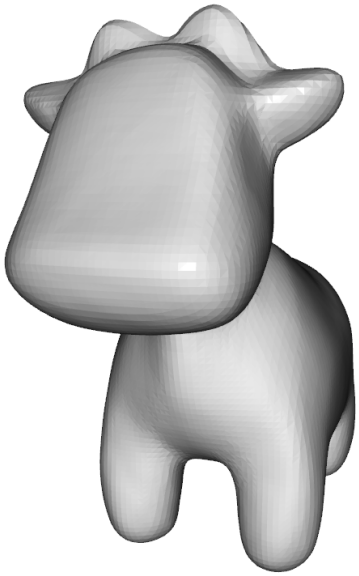
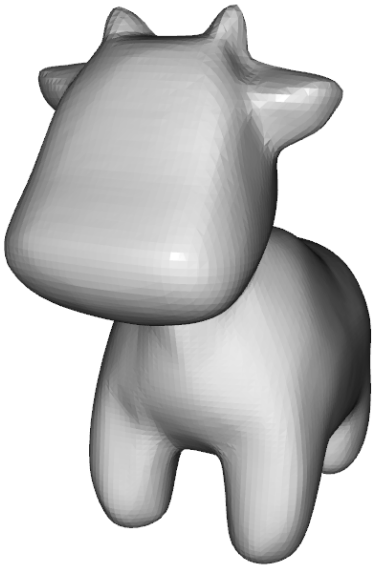
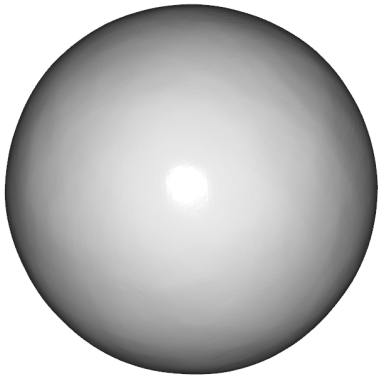
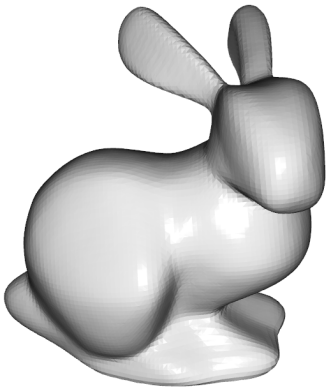
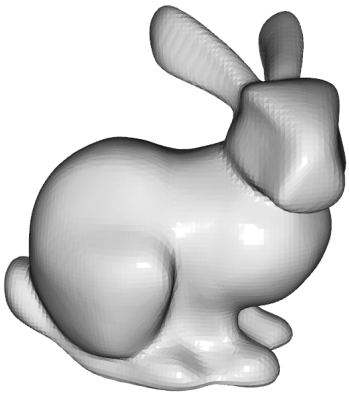
Rendered meshes extracted from the surface computed by the single-deepsdf model visualized with the mesh on which the model was trained (the mesh is interpreted as a point cloud for training). A blender subdivision surface-modifier is used to derive the point-cloud data, normals are computed by blender (original model is OBJ format so it only had verts and faces).

		
		
250 points (Base model) Best Epoch: 29 Loss $_{1}$: (.0396, .0385) Loss $_{29}$: (.0188, .0157) Train-time: ~20s	1500 points (1 subdivision) Best Epoch: 38 Loss $_{1}$: (.0376, .0321) Loss $_{38}$: (.0077, .0062) Train-time: ~100s	6000 points (2 subdivisions) Best Epoch: 97 Loss $_{1}$: (.0269, .0179) Loss $_{97}$: (.0031, .0022) Train-time: ~375s

It is clear that this method works best when the input point cloud has many good training values for detail-dense areas such as the horns/ears of the cow, but having higher-fidelity point data on relatively smooth areas like the jaw doesn't make much difference and is definitely not worth the impact on the time it takes to train the model.

There is also more variability in the resulting surface when there is less training data, (i.e. the leftmost cow will have very different head shapes from model to model), so maybe it would help to switch up the validation and training sets between training epochs.

Here are the requested images:

		
150 points	500 points	1000 points