Supplementary Material
Enabling Viewpoint Learning through Dynamic Label Generation

M. Schelling[1], P. Hermosilla[1], P.-P. Vázquez[2] and T. Ropinski[1]

[1]Ulm University, Germany
[2]Universitat Politècnica de Catalunya, Spain

Appendix A: Mesh Cleaning Pipeline

Our mesh cleaning pipeline consists of three steps. First we resolve self intersections of the mesh, using PyMesh. These are particularly bad as in this case the area of a face does not correspond to its potentially visible area, as parts of a face can be hidden inside the model, changing the values of A_z and A_t. As a second step we remove non-surface polygons by computing the visibility of the faces from 1000 views and drop all non-visible faces of the model using MeshLab. This is primarily done to create cleaner surface meshes by removing unwanted parts of the model, e.g. passenger seats inside planes. This results in a A_t being closer to the actual surface area of the model, while also speeding up the downstream tasks by reducing the number of polygons per model. As the first step introduces artifacts in the form of small and irregular meshing, where self-intersections were resolved, we add a third and last step where we regularize the surface meshes by performing an edge-collapse reduction algorithm, again using MeshLab. Furthermore, the last step also removes unwanted structures in the meshes, e.g. polygons referring to different textures, which are not relevant for shape information but can influence the viewpoint quality. Fig. 1 shows details of the model *airplane_0004* from ModelNet40, which contains self-intersections (top left) and unnecessary polygons (top right). The proposed mesh processing resolves self-intersections in the first step and and cleans the meshing in the third step (bottom images). We note that this mesh cleaning pipeline does not create perfect watertight surface meshes, but is merely a trade-off between computation time and achieved mesh quality. The resulting quality turned out to be adequate for our experiments. Providing an algorithm for high quality remeshing is beyond the scope of this paper, and an active field of research on its own.

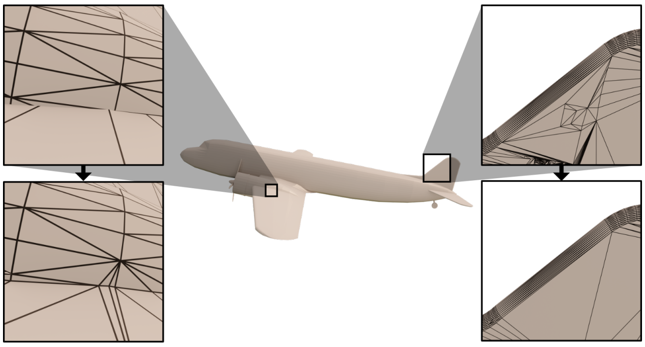

Figure 1: **Mesh cleaning.** Results of the different steps to clean the meshes on *airplane_0004*. The original mesh contains self intersections (left) and non-uniform meshing artifacts (right), which are resolved in the first and third step our mesh cleaning pipeline, respectively.

Appendix B: Distribution Learning

Network Architecture

For predicting the viewpoint quality distribution we use the same feature encoder as for the other tasks, see Section 4.3, and replace the prediction MLPs with 2D decoder networks. These decoder networks consists of deconvolution layers to increase the spatial dimensions interweaved with residual blocks to increase the decoder capacity. The whole decoder architecture is as follows:

- 2D deconvolution with filter size 4×4 and stride 1,
- 2 ResNet blocks with filter size 3×3 and depth 2 each,
- 2D deconvolution with filter size 4×4 and stride 4
- 2 ResNet blocks with filter size 3×3 and depth 2 each,
- 2D deconv layer with filter size 2×2 and stride 2,

with batch normalization and ReLU activation in between all layers. The respective spatial dimensions are $(1\times1, 4\times4, 16\times16, 32\times32)$ with feature dimensions $(2048, 1024, 256, 1)$.

Predicted Distributions

Examples for predicted viewpoint distributions can be seen in Fig. 2.

Appendix C: Viewpoint prediction

Tables 1 shows the test results on the different categories for the experiment from Section 5.2. Our combined ML+GL method achieves best performance on almost all categories and viewpoint quality measures. The results are comparable on all categories for all viewpoint quality measures, with or without considering statistics,

Table 1: **Detailed results.** Breakdown from Table 2 for each category.

		airplane	bench	bottle	car	chair	sofa	table	toilet	mean
VE	SL	60.7	50.9	64.8	58.0	63.8	88.7	38.3	74.4	62.4
	SR	49.4	65.1	53.3	64.0	66.3	63.4	73.5	70.0	63.1
	DLDL	47.0	55.4	53.4	58.7	62.9	63.4	56.5	72.4	58.7
	ML	55.4	62.1	50.4	79.8	**73.9**	83.3	76.4	79.2	70.1
	GL	70.0	**69.5**	52.9	82.0	71.7	87.9	76.6	83.4	74.2
	ML+GL	**79.1**	67.7	**75.3**	**84.0**	73.0	**88.8**	**83.0**	**83.8**	**79.3**
VR	SL	71.4	71.7	69.9	65.1	72.4	**76.4**	76.4	64.9	71.0
	SR	71.0	73.4	72.2	69.3	65.3	59.2	80.3	67.4	69.8
	DLDL	69.5	70.2	72.6	66.3	67.0	59.5	69.3	58.9	66.6
	ML	63.2	73.6	70.0	74.2	77.5	73.3	81.0	64.0	72.1
	GL	66.2	**83.0**	69.1	78.6	75.5	75.5	80.8	72.4	75.1
	ML+GL	**74.8**	72.8	**78.0**	**80.3**	**77.9**	75.7	**82.0**	**84.3**	**78.2**
VKL	SL	89.2	76.2	74.9	83.7	83.5	86.3	86.0	65.6	80.7
	SR	86.2	84.4	88.9	74.0	72.1	79.1	89.0	71.5	80.6
	DLDL	86.7	83.3	88.9	73.0	72.0	73.0	78.8	67.8	77.9
	ML	79.7	79.8	92.7	80.4	86.2	75.5	90.2	76.5	82.6
	GL	91.8	**88.1**	90.9	85.3	89.3	**94.0**	90.9	84.4	89.3
	ML+GL	**95.2**	85.5	**94.9**	**89.7**	**90.8**	92.2	**91.6**	**89.8**	**91.2**
VMI	SL	90.6	79.2	80.4	84.0	88.3	92.6	84.3	64.4	83.0
	SR	85.0	86.2	86.5	81.6	70.6	75.7	**91.1**	64.0	80.1
	DLDL	87.8	79.8	86.9	79.6	72.1	69.9	80.0	67.0	77.9
	ML	88.7	68.7	92.1	80.2	89.9	81.9	87.7	67.8	82.1
	GL	94.0	85.1	91.5	78.7	91.0	91.2	88.8	81.7	87.7
	ML+GL	**96.6**	**87.3**	**94.1**	**92.2**	**93.0**	**93.5**	90.1	**93.4**	**92.5**

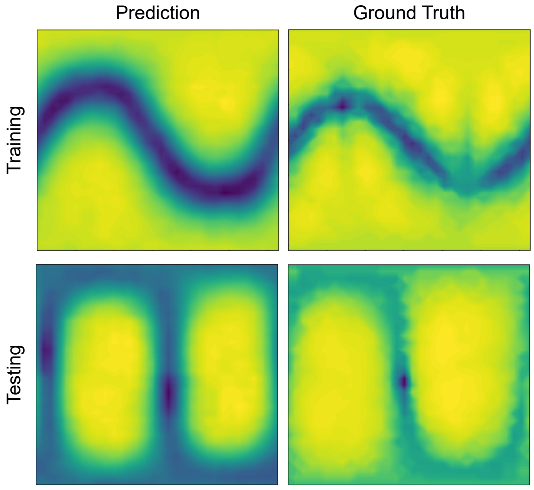

Figure 2: **Predicted viewpoint quality distributions.** Results of the DLDL approach on the category *airplane* for one example from the training set and one example from the test set.

Appendix D: Dataset

We release our training data which contains dense viewpoint quality values for 1k viewpoints on a Fibonacci sphere for VE, VR, VKLand VMI for ~12k models from ModelNet40. For a subset of ~4k models from the categories *airplane, bench, bottle, car, chair, sofa, table and toilet* we additionally provide the cleaned models using our pipeline together with the sampled viewpoint quality values for VE, VR, VKL and VMI.