Appendices

A. Edge Count Difference

The exact formula to compute the Edge Count Difference (ECD) [4] between sets $\mathcal{A}$ and $\mathcal{B}$, used as evaluation measure for all experiments.

$$\text{ECD} = (R_1 - \mu_1, R_2 - \mu_2)\Sigma^{-1}\begin{pmatrix} R_1 - \mu_1 \\ R_2 - \mu_2 \end{pmatrix} \quad (2)$$

R_1 and R_2 are the counted edges within $\mathcal{A}$ and $\mathcal{B}$ respectively. The expected values for R_1 and R_2 are given as μ_1 and μ_2. Σ is the covariance matrix of the vector (R_1, R_2) under the permutation null distribution. Specifically, that means

$$\mu_1 = |G|\frac{n(n-1)}{N(N-1)}$$

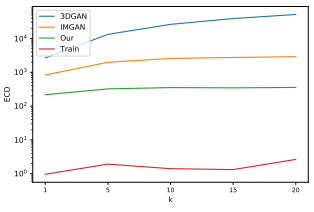
$$\mu_2 = |G|\frac{m(m-1)}{N(N-1)}$$

$$\Sigma_{11} = \mu_1(1-\mu_1) + 2C\frac{n(n-1)(n-2)}{N(N-1)(N-2)} + (|G|(|G|-1) - 2C)\frac{n(n-2)(n-2)(n-3)}{N(N-1)(N-2)(N-3)}$$

$$\Sigma_{22} = \mu_2(1-\mu_2) + 2C\frac{m(m-1)(m-2)}{N(N-1)(N-2)} + (|G|(|G|-1) - 2C)\frac{m(m-2)(m-2)(m-3)}{N(N-1)(N-2)(N-3)}$$

$$\Sigma_{12} = \Sigma_{21} = (|G|(|G|-1) - 2C) \cdot \frac{nm(n-1)(m-1)}{N(N-1)(N-2)(N-3)} - \mu_1\mu_2$$

where G is the k-MST build from set $\mathcal{A}$ and $\mathcal{B}$, $n = |\mathcal{A}|$, m = $|\mathcal{B}|$ and $N = m+n$. C is given as $C = \frac{1}{2}\sum_{i=1}^{N}|G_i|^2 - |G|$, with G_i being the subgraph in G that includes all edges that connect to node i. All formulas are from [4]. The only parameter of this method is k, the minimal number of neighbors of each vertex in the MST. Although this parameter does effect the magnitude of the score, relative distances do not change significantly, as we show on the right on the example of the chair dataset. For all of our experiments k is set to 10.



B. Network Architectures

In this section we go into detail on all architectures used in our experiments. In our figures a rectangle signifies data, with its given size. A rounded rectangle stands for layers of our networks. For convolutions we note the number of channels, the kernel size and the stride.

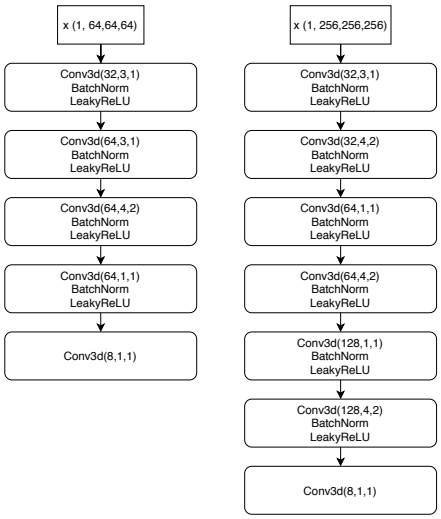


Figure 9: Two different encoders used for high or low input resolutions. In both cases the same decoder is used. The encoders receive as input a voxel grid with a resolution of 64 or 256 respectively. In both cases the output resolution is 32 and 8 channels are used

C. Training

All experiments were done on a GeForce RTX 2080 Ti. We used Adam [15] as optimizer and a learning rate of $1e-3$ for the autoencoder, generator and discriminator with no weight decay.

Autoencoder We trained the autoencoder for 200 epochs, although a lower number would probably suffice, as the AE converges fast to satisfactory results. For the high-resolution version we used a batch size of 16 for the low-resolution version of 8. Per object the implicit function was sampled at 6000 positions. For the high-resolution version the entire grid does not fit into memory, therefore we randomly carve a 3D slice of resolution 48 out for processing.

GAN We trained the GAN for 500 epochs with a batch size of 48. The gradient penalty weight was chosen as 1.

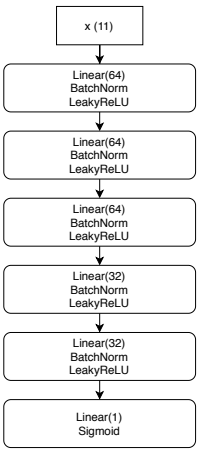


Figure 10: The decoder used in all our experiments. It gets as input a points coordinates relative to the cell center it is located in, concatenated with the cells latent vector. The output can be rounded to a binary value, telling us whether the point is inside our outside of the shape

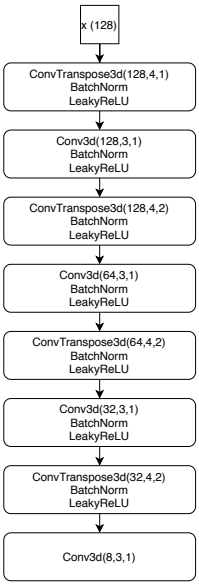


Figure 11: For unconditional generation we use a generator with a simple convolutional architecture.

The training time ranged from 20 hours for unconditional generation on the rifle dataset to 100 hours for conditional generation on the table dataset.

D. Evaluation

We conducted an ablation study, to show the effect of different choices regarding the discriminator (Table 4). When choosing a regular discriminator, instead of a patch-based architecture, we observe significant mode collapse. Furthermore, we show that each of the three used discriminators improves the training result. It should come as no surprise,

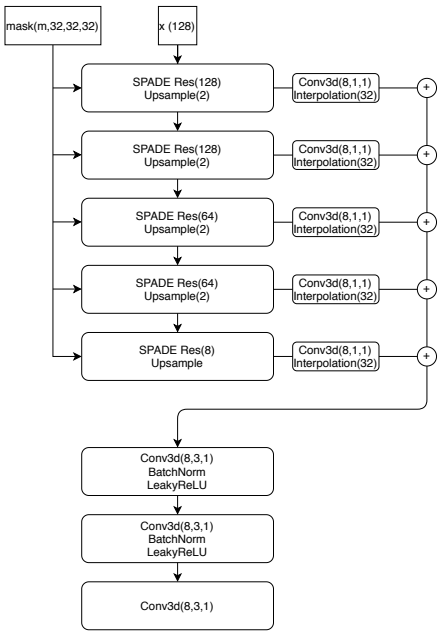


Figure 12: Our conditional generator is inspired by SPADE [27], where the mask is used to compute cell-wise scales and biases. The number of channels m depends on the application. For the design of the SPADE Res block we refer to [27]. We added skip connections to their architecture.

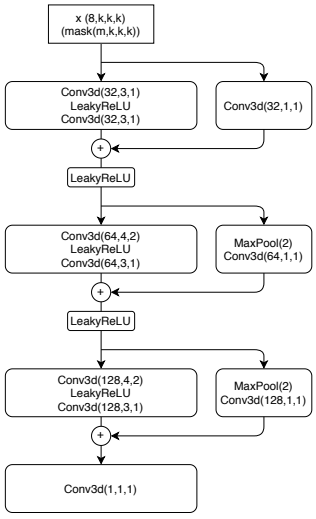


Figure 13: The patch discriminator used in all experiments. Spectral normalization is applied to all convolutional layers. The input resolution k is either 32, 16 or 8. For conditional generation the number of mask channels m depends on the application. The mask is simply concatenated to the latent grid.

that the discriminators at higher resolution are more important for the results.

	ECD	MMD	COV
no patch	27347	6818	0.00
16 8	6158	3265	74.71
32 8	323	2778	76.40
32 16	180	2784	81.49
complete	144	2768	82.09

Table 4: Ablation study on the chair dataset. We show results for a standard (not patch-based) discriminator. Furthermore, we show, that the results worsens, when leaving out one of the three discriminators. The best results are obtained when using all three.

As it is straightforward for our method to produce results in higher resolutions, we report numbers at a resolution of 256 as well (Table 5). For these comparisons we do not compute distances to voxelized ground truth meshes but to the original ones. Therefore these numbers are not comparable to our other results, but might be of interest for future comparisons. We furthermore report results for conditional generation. For this we conditioned on the bounding boxes obtained from the test set.

Furthermore, we show additional models generated with our approach both for unconditional (Figure 14) and bounding box based generation (Figure 15). The displayed objects are randomly sampled.

We further add a numerical evaluation of the bounding box fit. As the bounding box masks are discretized to a resolution of 32, we expect the difference between masks and actual bounding boxes to be between 0 and 1/32. As can be seen in Figure 16 the bounding boxes of most of our objects fall into this range.

Lastly, we demonstrate the effect smoothing has on the autoencoder results (Figure 17). When no smoothing is applied distinct borders between individual cells are visible.

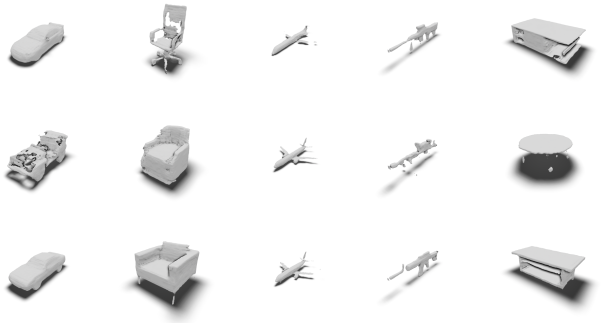


Figure 14: Results from our unconditional generator, sampled at random

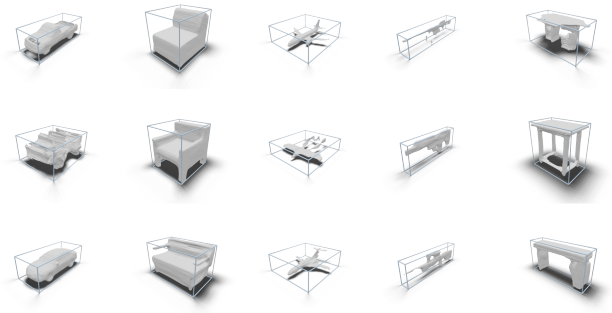


Figure 15: Results from our generator conditioned on bounding boxes, sampled at random

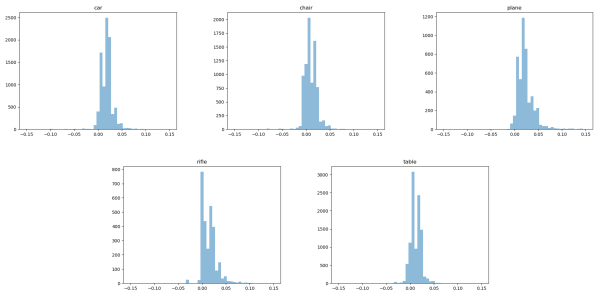


Figure 16: Numeric results to evaluate the fit of the shapes to their bounding boxes. The generation is conditioned on the bounding boxes of the test set. Note that we discretize the bounding boxes when using them as masks. Therefore errors between 0 an 1/32 are expected.

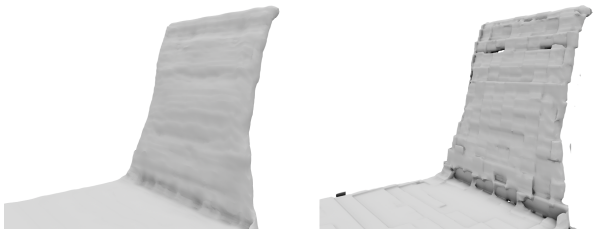


Figure 17: To demonstrate the effect of smoothing the classification results in a trilinear manner we show a generated chair with and without smoothing

		Plane	Car	Chair	Rifle	Table	Avg.
COV(%)	Unconditional	76.89	74.67	82.82	73.89	85.61	78.78
	Conditional	64.15	71.80	70.65	65.26	80.32	70.43
MMD	Unconditional	4,189	1,507	3,125	4,125	2,639	3,117
	Conditional	4,422	1,567	3,223	4,383	2,729	3,265
ECD	Unconditional	2,390	6,043	369	366	349	
	Conditional	2,394	8,057	1,270	413	649	

Table 5: Quantitative evaluation of our generative models at resolution 256 to the ground truth. For conditional generation we use the bounding boxes of the test set