Supplementary Materials for MRGAN: Multi-Rooted 3D Shape Representation Learning with Unsupervised Part Disentanglement

A. Additional results

In Figure 1 we show additional samples generated from our three classes. Figure 2 shows additional mixing results for the chair and table classes.

Note in particular that part mixing does not simply consist of copying the parts as-is, but that the network attempts to maintain their identity while preserving the overall coherence. This is particularly prominent in the second row of the chairs in Figure 2, where the round chair back is maintained, but its size and inclination are adjusted to fit with the rest of the chair. A similar effect can be observed for the table class, where the identity of legs (e.g. their inclination or curvature) is maintained, but the height and width of the resulting table are adjusted in order to avoid an imbalance.

B. Baseline Mixing Results

We provide qualitative evaluations of a shape-mixing experiment conducted using the unsupervised baseline model (BAE-Net [4] + CompoNet [7]). For each figure, we randomly generated 10 shapes by randomly sampling latent codes for each part and a code for shape composition (see [7] for details on their latent structure). We randomly split these shapes into two groups of five, and generated a pairwise mixing between the groups. For each such pair, we generate a novel shape by assigning the latent codes from one of the paired shapes to half the parts, and the latent codes from the other shape for the reminder. The composition code was similarly taken from one of the paired shapes.

Figure 3 shows the results of mixing on the table and airplane classes. The model often generates disconnected parts, and some components show virtually no variability (see for example the shape of the tails in the Airplane class). These observations are similarly reflected in the quantitative results of Table 3 of the core paper.

Figure 4 shows the results of mixing on the chair class. In this scenario, some parts are not common across all instances of the class, and the model fails to avoid the generation of redundant parts (both legs and a swivel base on a chair) or meaningfully maintain their identity when mixing shapes (armrests appearing when neither mixed shape has them). Our model, meanwhile, has no such limitations.

C. Comparisons on additional metrics

We provide comparisons results with standard generative frameworks on all metrics of [1] in Table 1. While our disentangled representation's quality metrics do not compete with the state of the art, they remain within the same order or improve upon some of the previous works.

We further compared the quality of our generated parts to the recent Mo et al. [6], using standard point cloud metrics and the PartNet dataset [9]. As our network does not learn a representation of specific parts (e.g. 'legs') we instead assigned to each class all the roots that contain points within that class. A root that contains both a table leg and a portion of the table top would therefore be included in both the legs and table top metrics.

This comparison shows that our self-learned inferred parts do not fall far behind the state of the art, even though the latter employs **full** part-level and hierarchy supervision.

Of note is the drop in coverage metrics between the shape-level and part-level comparisons. We believe these differences arise because the given object classes display considerably more variation at the full-shape level than at the part-level. As our supervision is only at the full-shape level, the network can reasonably fool the discriminator by learning to represent only a subset of commonly occurring part shapes and devoting more effort to composing them in varied ways.

D. Expanded ablation and mutual information

In addition to the meaningful distance metric, we evaluated the quality of disentanglement in our learned representation using a mutual information based score. For each scenario, we sampled 500 shapes and modified each of their roots individually, one at a time. For each such modification, we evaluated the empirical mutual information between the points sampled from each root (both modified and unmodified) before one of them was changed, and the points sampled from the same roots after the change. Mutual information was estimated using MINE [2]. Higher mutual information scores indicate strong dependence between the distribution of points before and after the change (e.g. if the points saw minimal change, or if they all underwent a simi-

Figure 1. Additional samples generated for the chair, table and airplane classes. Points are colored according to the root they originate from. We used 6 roots for the chairs and 5 roots for the tables and airplanes.

lar translation) while a lower score indicates the opposite.

In Table 3 we expand on the ablation results of the core paper, showing both the mutual information scores and two additional scenarios.

The full model exhibits the best results on both metrics, showing that it provides the best balance between the ability to affect meaningful changes in the modified part, while limiting alterations to the unmodified parts. Note in particular the sharp decline in both metrics when the convexity constraints (the hull-distance loss) are removed. In this scenario, the network conceals the information required to satisfy the reconstruction loss by inducing large displacements on a small subset of the points in each part. This drastically lowers the amount of change in the modified part, and leads to significant decrease in disentanglement.

Figure 2. Additional part mixing results for the chair and table classes. For each class, the shapes in the left column and the top row were generated from a single latent vector each. The rest of the shapes were generated by utilizing the latent vector from the top row for some of the roots, and the latent vector from the left column for the rest.

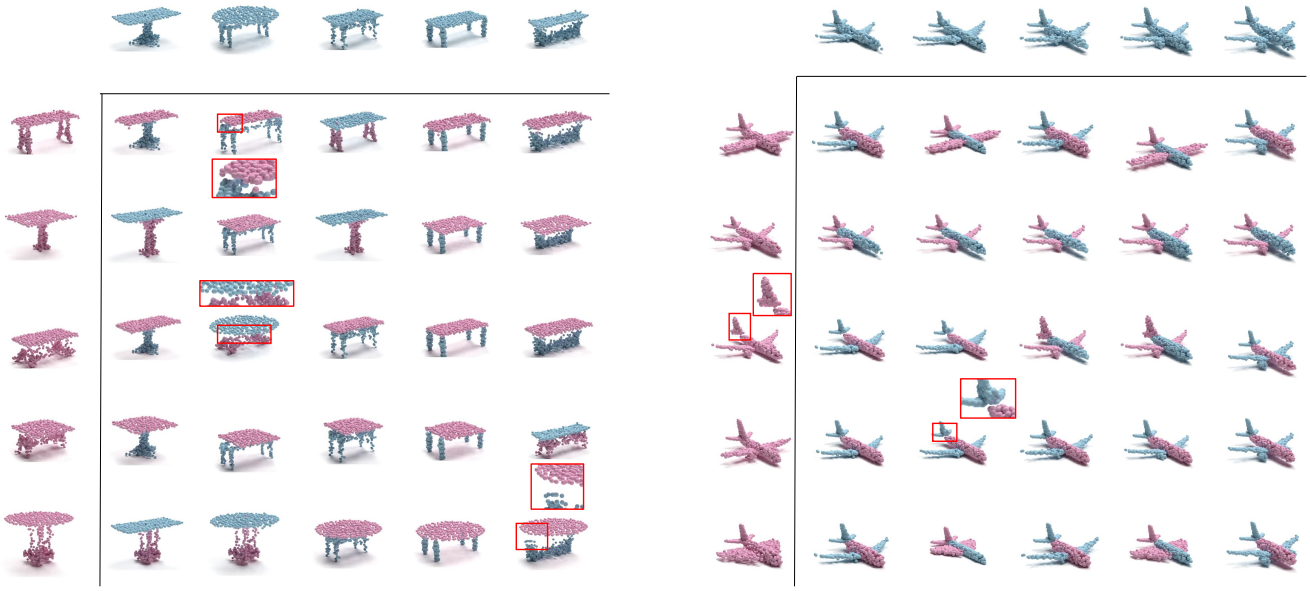

Figure 3. Baseline part mixing results for the table and airplane classes, using the combined BAE-Net + CompoNet model. For each class, the shapes in the left column and top row were generated from randomly sampled latent codes. The rest of the shapes were generated by utilizing the latent vectors from the top row for half the parts, and the latent vectors from the left column for the rest. Note that both classes contain disconnected parts (red) and display significantly diminished variation compared to our model (see for example the airplane tails).

E. Mixing strategies

We evaluated several root mixing strategies as part of our ablation study. A sample of the results is shown in Table 4, and a discussion thereof is provided in the paper's experimental section.

We note here again that the key observation from these experiments is that in order to achieve convergence along with a part-level disentanglement, we require a strategy that provides a gradient between the two scenarios at the extreme ends - those where a single latent code is fed into all

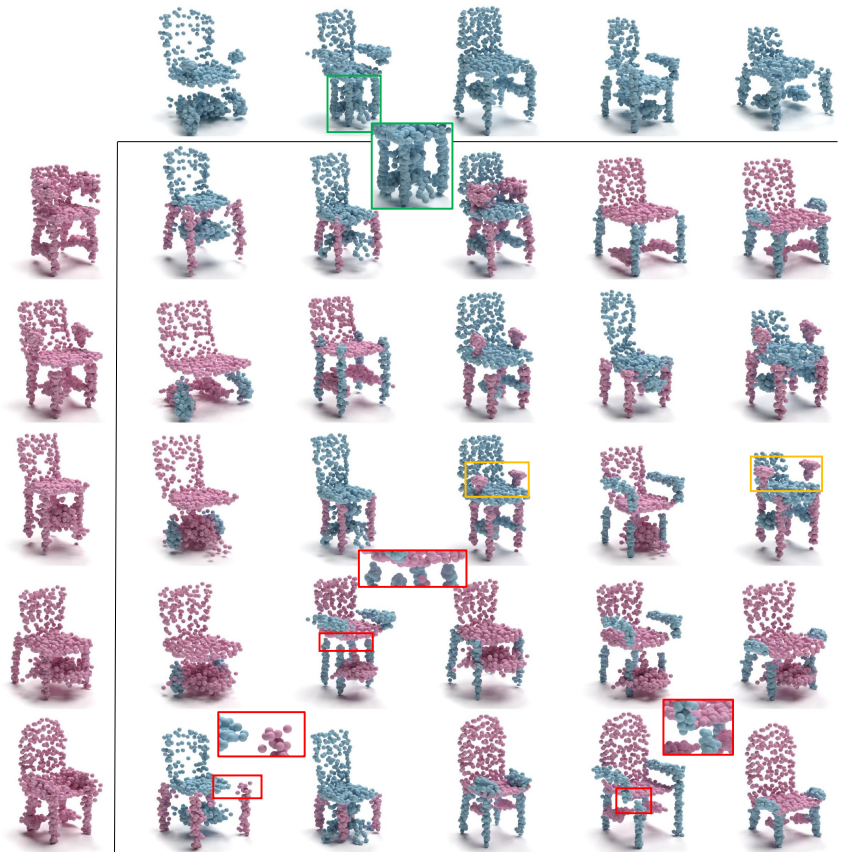

Figure 4. Baseline part mixing results for the chair class, using the combined BAE-Net + CompoNet model. Shapes were generated in a similar manner to Figure 3. The baseline is prone to generating disconnected parts (red), redundant parts (green) or adding parts that did not exist in either of the source shapes (yellow).

Class	Model	JSD ↓	MMD-CD ↓	MMD-EMD ↓	COV-CD ↑	COV-EMD ↑
Chair	r-GAN (dense)	0.238	0.0029	0.136	33	13
	r-GAN (conv)	0.517	0.0030	0.223	23	4
	Valsesia et al. (no up.)	0.119	0.0033	0.104	26	20
	Valsesia et al. (up.)	0.100	0.0029	0.097	30	26
	tree-GAN	0.119	0.0016	0.101	58	30
	MRGAN (ours)	0.246	0.0021	0.166	67	23
Airplane	r-GAN (dense)	0.182	0.0009	0.094	31	9
	r-GAN (conv)	0.350	0.0008	0.101	26	7
	Valsesia et al. (no up.)	0.164	0.0010	0.102	24	13
	Valsesia et al. (up.)	0.083	0.0008	0.071	31	14
	tree-GAN	0.097	0.0004	0.068	61	20
	MRGAN (ours)	0.243	0.0006	0.114	75	21
Table	tree-GAN	0.077	0.0018	0.082	71	48
	MRGAN (ours)	0.287	0.0020	0.155	78	31

Table 1. Comparisons to previous generative point cloud models on the metrics of [1]. We use the values reported by [8] where applicable. For the airplane and chair classes, the best and second best results are colored in red and blue respectively.

roots and where a different code is fed into each root.

We believe this balance is required, as providing too great a weight to the single-code scenario allows the network to learn that the different inputs are correlated and thus they can be ignored. On the other hand, providing this scenario allows the network to initially converge towards

Class	Part	Model	JSD ↓	MMD-CD ↓	MMD-EMD ↓	COV-CD ↑	COV-EMD ↑
Chair	Legs	PT2PC	0.069	0.068	0.320	50	49
		MRGAN (ours)	0.319	0.101	0.550	23	15
	Back	PT2PC	0.104	0.075	0.363	34	31
		MRGAN (ours)	0.333	0.161	0.419	12	13
	Seat	PT2PC	0.089	0.081	0.370	32	26
		MRGAN (ours)	0.168	0.138	0.392	13	14
Table	Legs	PT2PC	0.078	0.053	0.293	44	47
		MRGAN (ours)	0.448	0.117	0.608	19	13
	Top	PT2PC	0.188	0.096	0.448	22	16
		MRGAN (ours)	0.534	0.159	0.636	13	12

Table 2. Part-level comparisons to [6] on the metrics of [1].

Model	Meaningful distance			Mutual information		
	Modified part	Unmodified parts	Ratio ↑	Modified part	Unmodified parts	Ratio ↓
Chair	0.096	0.017	5.36	0.577	0.977	0.590
Table	0.080	0.029	2.71	0.438	0.820	0.535
Airplane	0.122	0.026	4.66	0.429	0.837	0.513
Airplane - Mix + GAN	0.107	0.050	2.13	0.735	1.139	0.646
+ Hull-distance loss	0.109	0.048	2.23	0.490	0.913	0.537
+ Triplet loss	0.123	0.050	2.46	0.466	0.815	0.572
+ Reconstruction loss	0.122	0.026	**4.66**	0.429	0.837	**0.513**
Full w/o Triplet loss	0.115	0.034	3.39	0.798	0.937	0.852
Full w/o Hull-distance loss	0.067	0.066	1.01	0.627	0.830	0.756

Table 3. Expanded ablation results, showing both the meaningful distance metric and the mutual information scores for modified and unmodified roots, as well as their ratio. In addition to the scenarios outlined in the core paper, we also present the scenarios where only the hull-distance and only the triplet loss were removed from the full model (below the dashed line).

the simpler root-ignoring minima, at which point the latent code reconstruction loss can drive the network to a solution that does not ignore the additional roots.

This behaviour is reminiscent of the mixing experiments in Karras et al. [5], where optimal separability scores were achieved for a scenario that contains 50% mixing.

For all strategies that contain a gradient between these two scenarios, the network successfully converges, reaching similar scores on our disentanglement metrics. These results indicate that the specific details of any chosen mixing strategy are less crucial.

F. Qualitative ablation

The effects of some of our losses are better appreciated visually. In Figure 5 we provide a visual contrast of their effects. Removal of the root-dropping loss (top row) results in the emergence of redundant parts (blue and orange roots). Removal of the convexity loss (bottom row) results in considerable degradation in the separation to meaningful parts.

We further provide a visualization of the disentanglement afforded by the network via two distance based mappings. For each scenario outlined in our ablation studies we show visualizations of both the meaningful distance metric presented in the paper, and traditional euclidean distance.

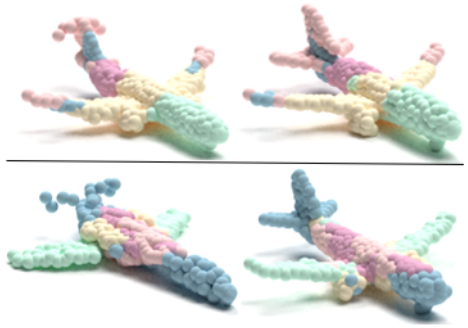

Figure 5. Loss removal effects. Top row: the root dropping loss was removed, leading to a redundancy between the orange and blue roots. Bottom row: the convexity loss was removed, leading to considerable degradation in separation into meaningful parts.

F.1. Meaningful distance visualization

Figure 7 shows a qualitative comparison of our ablation scenarios using the meaningful distance metric. For each row we consider a single generated shape and modify the latent code provided to one root at a time. Points that undergo a *meaningful* change, i.e. those that have moved a distance comparable to the mean change between randomly sampled shapes, are colored. If the points are part of the

Mixing Strategy	Meaningful distance			Mutual information		
	Modified part	Unmodified parts	Ratio ↑	Modified part	Unmodified parts	Ratio ↓
(same)			No disentanglement			
(different)			Does not converge			
(same) + (different)			No disentanglement			
(same) + (different) + (one)	0.135	0.030	4.53	0.446	0.853	0.522
(same) + (different) + (half)	0.150	0.034	4.44	0.453	0.871	0.520
(same) + (different) + (one) + (half)	0.122	0.026	4.66	0.429	0.837	0.513

Table 4. Meaningful change and mutual information metrics for the Airplane class utilizing different mixing strategies. (same) indicates the same input to all roots, (half) indicates a half-and-half mixing of 2 latent codes between the roots, (one) indicates mixing a single latent code for one random root and a second latent code for all other roots and (different) indicates a different latent code for each root. For each strategy, all scenarios are sampled uniformly.

Figure 6. Chairs produced from a varying root model. At training time, the model was allowed to uniformly sample a number of roots with which to generate each chair. At inference time, we manually choose a number of roots, sample a latent code for each and produce a novel chair. The choice of the number of roots dictates the type of generated shape - 4-legged (6 roots, left) or swivel chairs (8 roots, right). However, root identities are not maintained between the two types - with the teal and pink parts, for example, shifting from legs to parts of the seat.

modified root, they are colored in green. Otherwise they are colored in red.

We can observe the effect of the different losses on the generated shapes. The hull distance loss better localizes the roots in meaningful parts, but it does not sufficiently drive disentanglement, with some latent codes controlling multiple parts and others controlling none. The triplet loss aids in achieving part-control but the points of each part are poorly localized. Finally, combining all our losses yields results which are both well localized and lead to visible changes.

F.2. Traditional locality visualization

We provide an additional visualization of the disentanglement via a traditional distance-based mapping (Figure 8). In each image, we modify the latent code of a single root and color all points in the cloud according to their euclidean (L2) distance from their location in the source shape. Red indicates a greater shift in the coordinates of a point, while green indicates a smaller or no change.

This distance-based visualization offers us another indication that the results of modifying a single root are largely constrained to a single part. Some changes lead to a more global shift. In particular, the bottom right airplane had an overall translation upwards due of the addition of a wheel to the nose. Despite this overall shift, the identities of the other parts are largely unchanged.

G. Varying number of roots

In Figure 6 we provide qualitative results for the varying root experiment. In this experiment, we investigated the effects of allowing the model to uniformly sample the number of roots used to generate each point cloud. The chair class was chosen due to the greater intra-class variance in the number of convexes required to decompose each of its members. By manually choosing a number of roots at inference time, we gain an additional measure of control over the type of chair produced (swivel or 4-legged). However, root identities are not maintained between shapes with a varying number of roots, and control over individual parts is lost. Additional discussion is provided in the experiments section of the paper.

H. Feature sharing layer

The feature sharing layer is implemented by a series of operations. We first max-pool all feature vectors over the point dimension. The resulting maximal feature vector is passed through a dense layer and then concatenated back to the feature vector of each individual point. Finally, the features of each point are passed through a 1D convolutional operator (with weights shared between the different points), using a kernel size of 1 and a number of filters equal to the original dimension of point features.

A visualization of this process is provided in Figure 9. The layer dimensions are provided in Table 5.

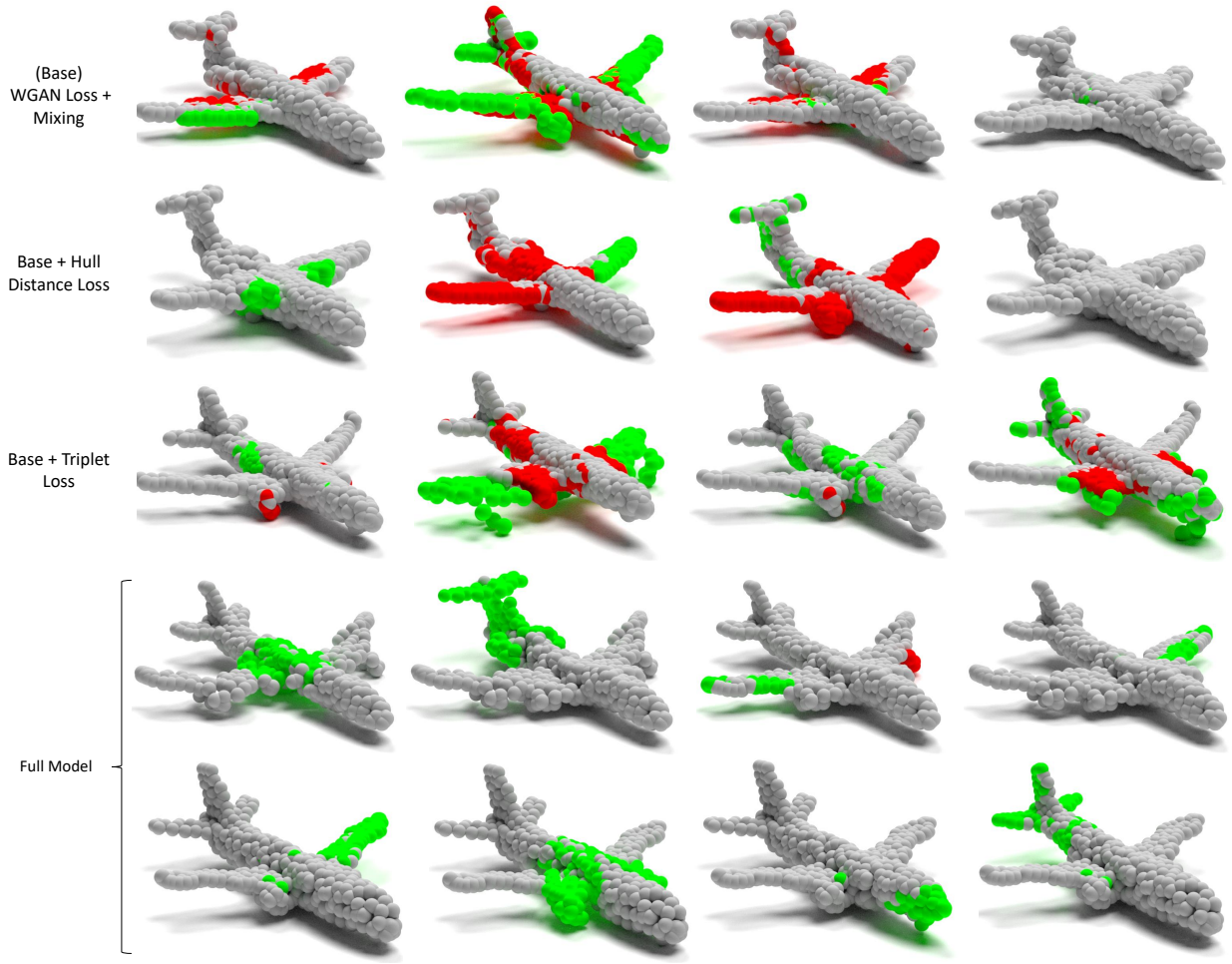

Figure 7. A visualization of root modification locality, using the meaningful distance metric. Each row displays one of our ablation scenarios. For each row we consider a single generated shape and modify the latent code provided to one root at a time. Points which have undergone a *meaningful* change are colored in green if they belong to the modified root and in red if they belonged to an unmodified root. Points which did not display a significant enough change are colored in grey. In the basic scenarios some modifications result in a complete change of identity (top row, second image from the left) or result in no visible change (second row, last image).

Layer type	Kernel size	Stride	Activation	Output dimension
MaxPool1D	P	1	-	1 x F
Dense	-	-	-	1 x 16
Concat	-	-	-	P x (F + 16)
Conv1D	1	1	LeakyReLU	P x F

Table 5. Network architecture of a feature sharing layer with P input points and F input features.

I. Network architecture

We describe the full network architecture for a generator with R roots in Table 6. The discriminator architecture is provided in Table 7. Table 8 provides the architecture details for the additional head used to reconstruct the initial latent codes from the discriminator features.

J. Hull-distance network details

The network architecture for our auxiliary distance-to-hull prediction network is given in Table 9. For pooling, we employ both a maximum pooling and a minimum pooling operation over the point dimension, and concatenate their results along the feature axis.

We train the network on a combination of sparse point samples from the ShapeNet [3] training set (including all available classes) as well as synthetic points sampled from a

Figure 8. A visualization of root modification locality, using an euclidean distance metric. Each image displays a heat map of individual point coordinate changes. Red indicates the point that had the largest shift while green indicates the smallest (or no) shift. While localization visibly improves for the full model, this metric is prone to over-representing changes borne from a more global shift. Note for example the wings on the bottom right plane, which have not changed their identity but display considerable movement due to an overall upwards shift resulting from the addition of a wheel to the nose.

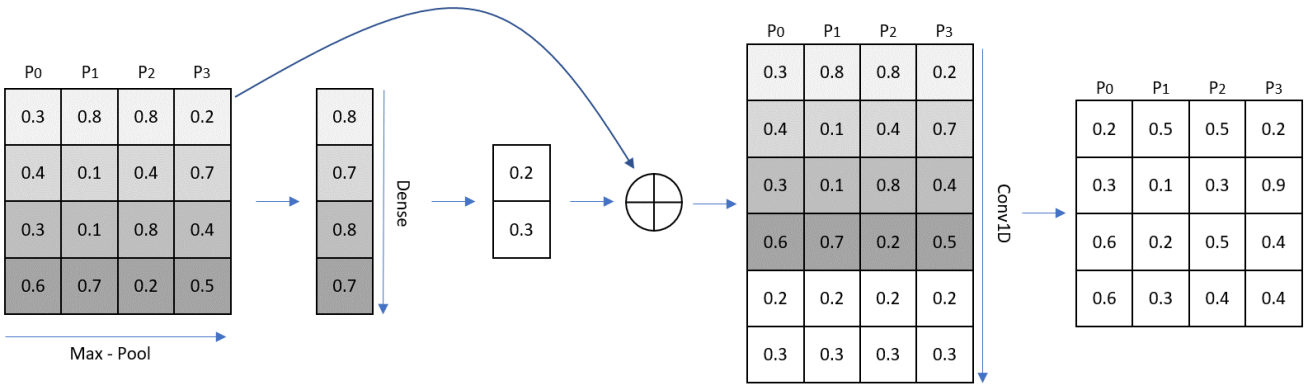

Figure 9. A visualization of the steps undertaken in the feature sharing layer.

sphere, a box, or any combination of the two with randomly sampled positions.

For each sampled cloud, we calculate the convex hull and compute the distance-to-hull metric analytically. The network is trained to predict this value for the input cloud, using an L_2 distance as the loss.

We employ an Adam optimizer with a learning rate of 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

The network is trained over 20,000 batches of 64 clouds each. In order to match the number of points in our genera-

Layer type	Branching ratio	Activation	Output dimension
Learned root constants	-	-	R x 256
TreeGCN	2	-	2R x 128
Feature sharing	-	LeakyReLU	2R x 128
TreeGCN	2	-	4R x 128
TreeGCN	2	-	8R x 128
TreeGCN	2	-	16R x 128
TreeGCN	16	LeakyReLU	256R x 3

Table 6. Network architecture of the multi-rooted generator with R roots. The root constants are normalized via an AdaIN layer, using a different latent vector as input for each root.

Layer type	Kernel size	Stride	Activation	Output dimension
Conv1D	1	1	LeakyReLU	256R x 3
Conv1D	1	1	LeakyReLU	256R x 64
Conv1D	1	1	LeakyReLU	256R x 128
Conv1D	1	1	LeakyReLU	256R x 512
Conv1D	1	1	LeakyReLU	256R x 1024
MaxPool1D	256R	1	-	1 x 1024
Dense	-	-	-	1 x 1024
Dense	-	-	-	1 x 512
Dense	-	-	-	1 x 512
Dense	-	-	-	1 x 1

Table 7. Network architecture of the discriminator, adapted from [1].

Layer type	Kernel size	Stride	Activation	Output dimension
Discriminator-Conv	-	-	LeakyReLU	256R x 1024
MaxPool1D	256	256	-	R x 1024
Conv1D	1	1	LeakyReLU	R x 512
Conv1D	1	1	LeakyReLU	R x 128
Conv1D	1	1	LeakyReLU	R x 128
Conv1D	1	1	Tanh	R x 96

Table 8. Network architecture of the discriminator's reconstruction head, used for the identity regularization term. The network begins from the outputs of the last convolutional layer of the discriminator.

Layer type	Kernel size	Stride	Activation	Output dimension
Conv1D	1	1	LeakyReLU	N x 3
Conv1D	1	1	LeakyReLU	N x 64
Conv1D	1	1	LeakyReLU	N x 128
Conv1D	1	1	LeakyReLU	N x 256
Conv1D	1	1	LeakyReLU	N x 512
Pooling	N	1	-	1 x 1024
Dense	-	-	LeakyReLU	1 x 512
Dense	-	-	LeakyReLU	1 x 256
Dense	-	-	LeakyReLU	1 x 128
Dense	-	-	LeakyReLU	1 x 64
Dense	-	-	LeakyReLU	1 x 31
Dense	-	-	LeakyReLU	1 x 1

Table 9. Network architecture of the auxiliary hull-distance prediction network. The pooling operation is the concatenation of both a min-pool and a max-pool operation employed along the point dimension.

tor's paths, we sample 256 points per cloud.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning Representations and Generative Models for 3D Point Clouds. In *ICLR*, 2018. 1, 4, 5, 9

[2] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning*, pages 531–540. PMLR, 2018. 1

[3] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 7

[4] Zhiqin Chen, Kangxue Yin, Matt Fisher, Siddhartha Chaudhuri, and Hao Zhang. BAE-NET: Branched autoencoder for shape co-segmentation. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 2019. 1

[5] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 5

[6] Kaichun Mo, He Wang, Xinchen Yan, and Leonidas J. Guibas. PT2PC: Learning to generate 3d point cloud shapes from part tree conditions. In *ECCV*, 2020. 1, 5

[7] Nadav Schor, Oren Katzir, Hao Zhang, and Daniel Cohen-Or. CompoNet: Learning to Generate the Unseen by Part Synthesis and Composition. In *ICCV*, pages 8758–8767, Seoul, Korea (South), 2019. 1

[8] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3D Point Cloud Generative Adversarial Network Based on Tree Structured Graph Convolutions. In *ICCV*, 2019. 4

[9] Fenggen Yu, Kun Liu, Yan Zhang, Chenyang Zhu, and Kai Xu. Partnet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation. In *CVPR*, page to appear, 2019. 1