Composite Shape Modeling via Latent Space Factorization

Anastasia Dubrovina[1] Fei Xia[1] Panos Achlioptas[1] Mira Shalah[1]
Raphaël Groscot[2] Leonidas Guibas[1]
[1]Stanford University [2]PSL Research University

Abstract

We present a novel neural network architecture, termed Decomposer-Composer, for semantic structure-aware 3D shape modeling. Our method utilizes an auto-encoder-based pipeline, and produces a novel factorized shape latent space, where the semantic structure of the shape collection translates into a data-dependent sub-space factorization, and where shape composition and decomposition become simple linear operations on the embedding coordinates. We further propose to model shape assembly using an explicit learned part deformation module, which utilizes a 3D spatial transformer network to perform an in-network volumetric grid deformation, and which allows us to train the whole system end-to-end. The resulting network allows us to perform part-level shape manipulation, unattainable by existing approaches. Our extensive ablation study, comparison to baseline methods and qualitative analysis demonstrate the improved performance of the proposed method.

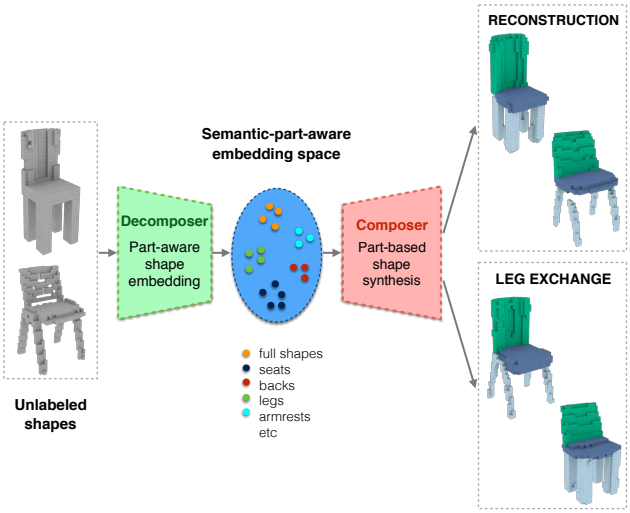

Figure 1: Given unlabeled shapes, the Decomposer maps them into a factorized latent space. The Composer can either reconstruct the shapes with semantic part labels, or create new shapes , for instance, by exchanging chair legs.

1. Introduction

Understanding, modeling and manipulating 3D objects are areas of great interest to the vision and graphics communities, and have been gaining increasing popularity in recent years. Examples of related applications include semantic segmentation [47], shape synthesis [41, 2], 3D reconstruction [8, 9], view synthesis [45], and fine-grained shape categorization [3], to name a few. The advancement of deep learning techniques, and the creation of large-scale 3D shape datasets [6] enabled researchers to learn task-specific representations directly from the existing data, and led to significant progress in all the aforementioned areas.

There is a growing interest in learning shape modeling and synthesis in a structure-aware manner, for instance, at the level of semantic shape parts. This poses several challenges compared to approaches considering the shapes as a whole. Semantic shape structure and shape part geometry are usually interdependent, and relations between the two must be implicitly or explicitly modeled and learned by the system. Examples of such structure-aware shape representation-learning are [24, 20, 39, 43].

However, the existing approaches for shape modeling, while being part aware at the intermediate stages of the system, still ultimately operate on the low-dimensional representations of the *whole* shape. For example, [24, 39] use a Variational Autoencoder (VAE) [16] to learn generative part-aware models of man-made shapes, but the latent spaces of the VAEs correspond to complete shapes, with entangled latent factors corresponding to different semantic parts. Therefore, these and other existing approaches cannot perform part-level shape manipulation, such as single part replacement, part interpolation, or part-level shape synthesis.

Inspired by the recent efforts in image modeling to separate different image formation factors, to gain better control over image generation process and simplify editing tasks [29, 35, 36], we propose a new semantic structure-

aware shape modeling system. This system utilizes an auto-encoder-based pipeline, and produces a factorized latent space which both reflects the semantic part structure of the shapes in the dataset, and compactly encodes different semantic parts' geometry. In this latent space, different semantic part embedding coordinates lie in *separate linear subspaces*, and shape composition can naturally be performed by summing up part embedding coordinates. The latent space factorization is data-dependent, and is performed using learned linear projection operators. Furthermore, the proposed system operates on *unlabeled* input shapes, and at test time it simultaneously infers the shape's semantic structure and compactly encodes its geometry.

Towards that end, we propose a Decomposer-Composer pipeline, schematically illustrated in Figure 1. The Decomposer maps an input shape, represented by an occupancy grid, into the factorized latent space described above. The Composer reconstructs a shape with semantic part-labels from a set of part-embedding coordinates. It explicitly learns the set of transformations to be applied to the parts, so that together they form a semantically and geometrically plausible shape. In order to learn and apply those part transformations, we employ a 3D variant of the Spatial Transformer Network (STN) [12]. 3D STN was previously utilized to scale and translate objects represented as 3D occupancy grids in [11], but to the best of our knowledge, ours is the first approach suggesting an in-network affine deformation of occupancy grids.

Finally, to promote part-based shape manipulation, such as part replacement, part interpolation, or shape synthesis from arbitrary parts, we employ the cycle consistency constraint [48, 29, 25, 38]. We utilize the fact that the Decomposer maps input shapes into a factorized embedding space, making it possible to control which parts are passed to the Composer for reconstruction. Given a batch of input shapes, we apply our Decomposer-Composer network twice, while randomly mixing part embedding coordinates before the first Composer application, and then de-mixing them into their original positions before the second Composer application. The resulting shapes are required to be as similar as possible to the original shapes, using a cycle consistency loss.

Main contributions Our main contributions are: (1) A novel latent space factorization approach which enables performing shape structure manipulation using linear operations directly in the learned latent space; (2) The application of a 3D STN to perform in-network affine shape deformation, for end-to-end training and improved reconstruction accuracy; (3) The incorporation of a cycle consistency loss for improved reconstruction quality.

2. Related work

Learning-based shape synthesis Learning-based methods have been used for automatic synthesis of shapes from complex real-world domains; In a seminal work [13], Kalogerakis *et al*. used a probabilistic model, which learned both continuous geometric features and discrete component structure, for component-based shape synthesis and novel shape generation. The development of deep neural networks enabled learning high-dimensional features more easily; 3DGAN [41] uses 3D decoders and a GAN to generate voxelized shapes. A similar approach has been applied to 3D point clouds and achieved high fidelity and diversity in shape synthesis [2].

Apart from generating shapes using a latent representation, some methods generate shapes from a latent representation with *structure*. SSGAN [40] generate the shape and texture for a 3D scene in a 2-stage manner. GRASS [20] generate shapes in two stages: first, by generating orientated bounding boxes, and then a detailed geometry within those bounding boxes. Nash and Williams [24] use use point cloud shape representation and a VAE to learn a probabilistic latent space of shapes; however, they require all training data to be in point-to-point correspondence. In a related work [39], Wang *et al*. introduced a 3D GAN-based generative model for 3D shapes, which produced segmented and labeled into parts shapes. Unlike the latter approach, our network does not use predefined subspaces for part embedding, but learns to project the latent code of the entire shape to the subspaces corresponding to codes of different parts.

In concurrent efforts, several deep architectures for part based shape synthesis were proposed [32, 19, 44, 23]. Schor *et al*. [32] utilized point-base shape representation, while operating on input models with known per-point parts labels. Li *et al*. [19] and [44] proposed two generative networks for part-based shape synthesis, operating on labeled voxelized shapes. Mo *et al*. [23] introduced a hierarchical graph network for learning structure-aware shape generation.

Spatial transformer networks Spatial transformer networks (STN) [12] allow to easily incorporate deformations into a learning pipeline. Kurenkov *et al*. [17] retrieve a 3D model from one RGB image and generate a deformation field to modify it. Kanazawa *et al*. [14] model articulated or soft objects with a template shape and deformations. Lin *et al*. [21] use STNs iteratively, to warp a foreground onto a background, and use a GAN to constrain the composition results to the natural image manifold. Hu *et al*. [11] use a 3D STN to scale and translate objects given as volumetric grids, as a part of scene generation network. Inspired by this line of work, we incorporate an affine transformation module into our network. This way, the generation module only needs to generate normalized parts, and the deforma-

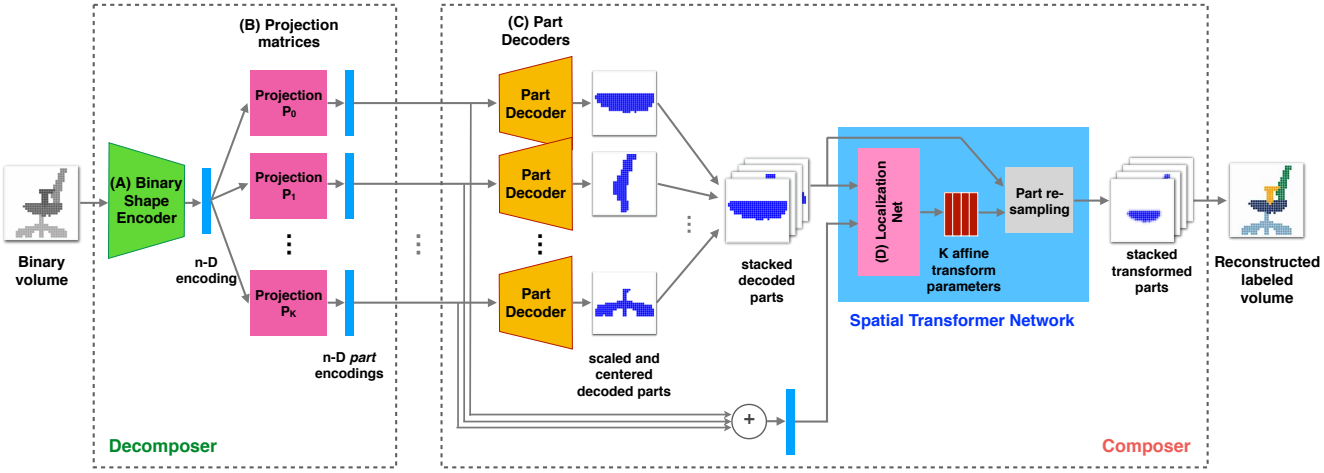

Figure 2: The proposed Decomposer-Composer architecture.

tion module transforms and assembles the parts together.

Deep latent space factorization Several approaches suggested to learn disentangled latent spaces for image representation and manipulation. β-VAE [10] introduce an adjustable hyperparameter β that balances latent channel capacity and independence constraints with reconstruction accuracy. InfoGAN [7] achieves the disentangling of factors by maximizing the mutual information between certain channels of latent code and image labels. Some approaches disentangle the image generation process using intrinsic decomposition, such as albedo and shading [36], or normalized shape and deformation grid [29, 35]. The proposed approach differs from [29, 35, 36] in that it maps both full and partial shapes into the same low dimensional embedding space, while in [29, 35, 36], different components have their own separated embedding spaces.

Projection in neural networks Projection is widely used in representation learning. It can be used for transformation from one domain to another domain [4, 27, 28], which is useful for tasks like translation in natural language processing. For example, Senel *et al.* [33] use projections to map word vectors into semantic categories. In this work, we use a projection layer to transform a whole shape embedding into semantic part embeddings.

3. Our model

3.1. Decomposer network

The Decomposer network is trained to embed unlabeled shapes into a factorized embedding space, reflecting the shared semantic structure of the shape collection. To allow for composite shape synthesis, the embedding space has to satisfy the following two properties: factorization consistency across input shapes, and existence of a simple shape composition operator to combine latent representations of different semantic factors. We propose to model this embedding space V as a *direct sum of subspaces* $\{V_i\}_{i=1}^K$, where K is the number of semantic parts, and each subspace $\{V_i\}$ corresponds to a semantic part i, thus satisfying the factorization consistency property. The second property is ensured by the fact that every vector $v \in V$ is given by a sum of unique $v_i \in V_i$ such that $V = V_1 \oplus ... \oplus V_k$, and part composition may be performed by part embedding summation. This also implies that the decomposition and composition operations in the embedding space are fully reversible.

A simple approach for such factorization is to split the dimensions of the n-dimensional embedding space into K coordinate groups, each group representing a certain semantic part-embedding. In this case, the full shape embedding is a concatenation of part embeddings, an approach explored in [39]. This, however, puts a hard constraint on the dimensionality of part embeddings, and thus also on the representation capacity of each part embedding subspace. Given that different semantic parts may have different geometric complexities, this factorization may be sub-optimal.

Instead, we perform a data-driven learned factorization of the embedding space into semantic subspaces. We use *learned* part-specific projection matrices, denoted by $\{P_i\}_{i=1}^K \in \mathbb{R}^{n \times n}$. To ensure that the aforementioned two factorization properties hold, the projection matrices must form a *partition of the identity* and satisfy the following three properties

$$\begin{aligned} &(1)\ P_i^2 = P_i, \forall i, \\ &(2)\ P_i P_j = 0 \text{ whenever } i \neq j, \\ &(3)\ P_1 + ... + P_K = I, \end{aligned} \tag{1}$$

where 0 and I are the all-zero and the identity matrices of size $n \times n$, respectively.

In practice, we efficiently implement the projection operators using fully connected layers without added biases,

with a total of $K * n^2$ variables, constrained as in Equation 1. The projection layers receive as input a whole shape encoding, which is produced by a 3D convolutional shape encoder. The parameters of the shape encoder and the projection layers are learned simultaneously. The resulting architecture of the Decomposer network is schematically described in Figure 2, and a detailed description of the shape encoder and the projection layer architecture is given in the supplementary material.

3.2. Composer network

The composer network is trained to reconstruct shapes with semantic part labels from *sets* of semantic part embedding coordinates. The simplest composer implementation would consist of a single decoder mirroring the whole binary shape encoder (see Figure 2), producing a semantically labelled reconstructed output shape. Such approach was used in [39], for instance. However, this straightforward method is known to fail in reconstructing thin volumetric shape parts and other fine shape details. To address this issue, we use a different approach, where we first separately reconstruct scaled and centered shape parts, using a *shared part decoder*. We then produce *per-part transformation parameters* and use them to deform the parts in a coherent manner, to obtain a complete reconstructed shape.

In our model, we make the simplifying assumption that it is possible to combine a given set of parts into a plausible shape by transforming them with per-part affine transformations and translations. While the true set of transformations which produce plausible shapes is significantly larger and more complex, our experiments demonstrate that the proposed simplified model is successful at producing geometrically and visually plausible results. This in-network part transformation is implemented using a 3D spatial transformer network (STN) [12]. It consists of a localization net, which produces a set of 12-dimensional affine transformations (including translations) for all parts, and a re-sampling unit, which transforms and places the reconstructed part volumes at their correct locations in the full shape. The SNT receives as input both the reconstructed parts from the part decoder, and the sum of part encodings, for best reconstruction results. The resulting Composer architecture is schematically described in Figure 2; its detailed description is given in the supplementary material.

We note that the proposed approach is related to the two-stage shape synthesis approach of [20], in which a GAN is first used to synthesize oriented bounding boxes for different parts, and then the part geometry is created per bounding box using a separate part decoder. Our approach is similar, yet it works in a reversed order. Namely, we first reconstruct part geometry, and then compute per-part affine transformation parameters, which are a 12-dimensional equivalent of the oriented part bounding boxes in [20]. Similarly

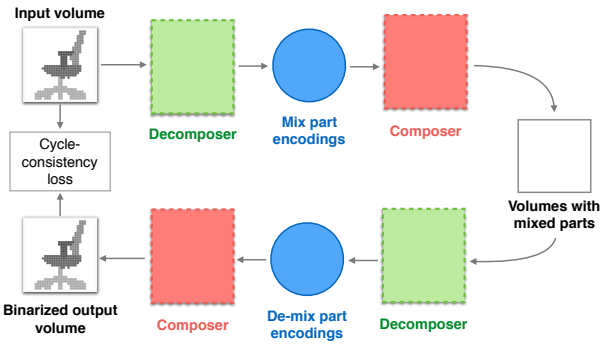

Figure 3: Schematic description of the cycle consistency constraint. See Section 3.3 for details.

to [20], this two stage approach improves the reconstruction of fine geometric details. However, unlike [20], where the GAN and the part decoder where trained separately, in our approach the two stages belong to the same reconstruction pipeline, trained simultaneously and end-to-end.

3.3. Cycle consistency

Our training set is comprised of 3D shapes with ground-truth semantic part-decomposition; It does not include any training examples of synthesized composite shapes. Existing methods for such shape assembly task operate on 3D meshes with very precise segmentations, and often with additional knowledge about part connectivity [46, 34, 13]. These methods cannot be applied to a dataset like ours, to produce a sufficiently large set of plausible new shapes (constructed from existing parts) to use for training a deep network for composite shape modelling. In order to circumvent this difficulty, and train the net to produce non-trivial part transformations for geometrically and semantically plausible part arrangements, we use a *cycle consistency* constraint. It has been previously utilized in geometry processing [25], image segmentation [38], and more recently in neural image transformation [29, 48].

Specifically, given a batch of M training shapes $\{X\}_{i=1}^{M}$, we map them to the factored latent space using the Decomposer, producing K semantic part encodings per input shape. We the randomly mix the part encodings of the shapes in the batch, while ensuring that after the mixing each of the new M encoding sets includes exactly one embedding coordinate per semantic part. We then reconstruct the shapes with correspondingly mixed parts using the Composer. After that, these new shapes are passed to the Decomposer-Composer pipeline once again, while demixing part encodings produced by the second Decomposer application, to re-store the original encoding-to-shape association. The cycle consistency requirement means that the final shapes are as similar as possible to the original M training shapes. We enforce it using the cycle consistency

loss described in the next section. The double application of the proposed network with part encoding mixing and demixing is schematically described in Figure 3.

3.4. Loss function

Our loss function is defined as the following weighted sum of several loss terms

$$L = w_{\text{PI}}\mathcal{L}_{\text{PI}} + w_{\text{part}}\mathcal{L}_{\text{part}} + w_{\text{trans}}\mathcal{L}_{\text{trans}} + w_{\text{cycle}}\mathcal{L}_{\text{cycle}}. \quad (2)$$

The weights compensate for the different scales of the loss terms, and reflect their relative importance.

Partition of the identity loss $\mathcal{L}_{\text{PI}}$ measures the deviation of the predicted projection matrices from the optimal projections, as given by Equation 1.

$$\mathcal{L}_{\text{proj}}(P_1, ..., P_k) = \sum_{i=1}^{K} \|P_i^2 - P_i\|_F^2 + \sum_{\substack{i,j=1,\\ i\neq j}}^{K} \|P_i P_j\|_F^2 + \|P_1 + ...P_K - I\|_F^2. \quad (3)$$

Part reconstruction loss $\mathcal{L}_{\text{part}}$ is the binary cross-entropy loss between the reconstructed centered and scaled part volumes and their respective ground truth part indicator volumes, summed over K parts.

Transformation parameter loss $\mathcal{L}_{\text{trans}}$ is an $L2$ regression loss between the predicted and the ground truth 12-dimensional transformation parameter vectors, summed over K parts. Unlike in the original STN approach [12], we found that direct supervision over the transformation parameters is critical for our network convergence.

Cycle consistency loss $\mathcal{L}_{\text{cycle}}$ is a binary cross-entropy loss between ground truth input volumes and their reconstructions, obtained using two applications of the proposed network, as described in Section 3.3.

3.5. Training details

The network was implemented in TensorFlow [1], and trained for 500 epochs with batch size 32. We used Adam optimizer [15] with learning rate 0.0001, decay rate of 0.8, and decay step size of 40 epochs. We found it was essential to first pre-train the binary shape encoder, projection layer and part decoder parameters separately for 150 epochs, by minimizing the part reconstruction and the partition of the identity losses and using $w_{\text{trans}} = w_{\text{cycle}} \approx 0$, for improved part reconstruction results. We then train the parameters of the spatial transformer network for another 100 epochs, while keeping the rest of the parameters fixed. After that we resume the training with all parameters and the cycle consistency loss to fine-tune the network parameters. The optimal loss combination weights were empirically detected using the validation set, and set to be $w_{\text{PI}} = 0.1, w_{\text{part}} = 100, w_{\text{trans}} = 0.1, w_{\text{cycle}} = 0.1$. The network was trained on each shape category separately.

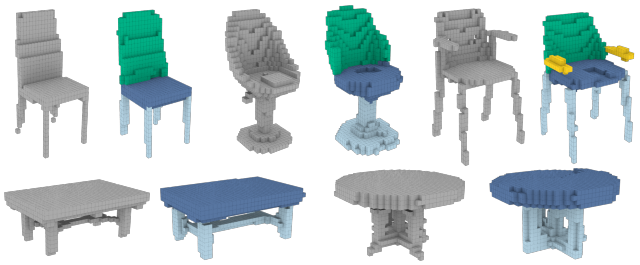

Figure 4: Reconstruction results of the proposed pipeline, for chair and table shapes. Gray shapes are the input test shapes; the results are colored according to the part label.

4. Experiments

Dataset In our experiments, we used the models from the ShapeNet 3D data collection [6], with part annotations produced by Yi *et al*. [47]. The shapes were converted to $32 \times 32 \times 32$ occupancy grids using binvox [26]. Semantic part labels were first assigned to the occupied voxels according to the proximity to the labeled 3D points, and the final voxel labels were obtained using graph-cuts in the voxel domain [5]. We used the official ShapeNet train, validation and test data splits in all our experiments. Additional results for $64 \times 64 \times 64$ occupancy grids can be found in the supplementary material.

4.1. Shape reconstruction

Figure 4 presents the results of reconstructing semantically labeled shapes from unlabelled input shapes, using the proposed network. Note that since our method performs separate part reconstruction with part decoders and part placement with an STN, it may produce less accurate part reconstruction, as compared to segmentation approaches - for example, the handles of the reconstructed rightmost chair in Figure 4. But, as illustrated by our quantitative study in Section 4.4, this allows us to perform better part-based shape manipulation.

4.2. Composite shape synthesis

Shape composition by part exchange In this experiment, we used our structured latent space to randomly swap corresponding embedding coordinates of pairs of input shapes (*e.g*., embedding coordinates of legs or seats of two chairs), and reconstruct the new shapes using the Composer. The results are shown in Figure 5, and demonstrate the ability of our system to perform accurate part exchange, while deforming the geometry of both the new and the existing parts to obtain a plausible result. See the supplementary material for additional results using four shape classes.

Shape composition by random part assembly In this experiment we tested the ability of the proposed network to assemble shapes from random parts using our factorized la-

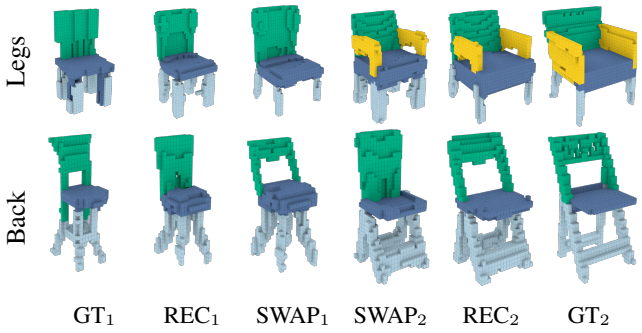

Figure 5: Single part exchange experiment. $GT_{1/2}$ denote ground truth shapes, $REC_{1/2}$ - reconstruction results, $SWAP_{1/2}$ - part exchange results. Unlabeled shapes were used as an input.

Figure 6: Shape composition by random part assembly. The top row shows the ground truth (GT) shapes, and the bottom row - shapes assembled using the proposed approach (see Section 4.2). Unlabeled shapes were used as an input.

tent space. Specifically, we mapped batches of input shapes into the latent space using the Decomposer, and created new shapes by randomly mixing the part embedding coordinates of the shapes in the batch, and reconstructing new shapes using the Composer. The results are shown in Figure 6, for chairs and tables, and illustrate the ability of the proposed method to combine parts from different shapes, scale and translate them so that the resulting shape looks realistic. See the supplementary material for additional shape composition results.

Full and partial interpolation in the embedding space In this experiment, we tested reconstruction from linearly interpolated embedding coordinates of complete shapes, as well as of a single semantic part. For the latter, we performed the part exchange experiment, described above, and interpolated the coordinates of that part, while keeping the rest of part embedding coordinates fixed. The results are shown in Figure 7. See the supplementary material additional interpolation results.

4.3. Latent space and projection matrix analysis

The latent space obtained using the proposed method exhibits clear separation into subspaces corresponding to different semantic parts. The projection matrices, while not not being strictly orthogonal, as required for the partition of the identity (1), have low effective ranks, which is in line with the clear separation into non-overlapping subspaces produced by them. See the supplementary material for the latent space and the projection matrices visualization.

4.4. Ablation study and comparison with existing approaches

4.4.1 Ablation study

To highlight the importance of the different elements of our approach, we conducted an ablation study, where we used several variants of the proposed method, listed below.

Fixed projection matrices Instead of using learned projection matrices in the Decomposer, the n-dimensional shape encoding is split into K consecutive equal-sized segments, which correspond to different part embedding subspaces. This is equivalent to using constant projection matrices, where the elements of the rows corresponding to a particular embedding space dimensions are 1, and the rest of the elements are 0.

Composer without STN We substituted the proposed composer, consisting of the part decoder and the STN, with a single decoder producing a labeled shape. The decoder receives the sum of part encodings as an input, processes it with two FC layers to combine information from different parts, and then reconstructs a shape with parts labels using a series of deconvolution steps, similar to the part decoder in the proposed architecture.

Without cycle loss We removed the cycle loss component during the network training.

4.4.2 Comparison with existing methods

Most existing methods for composite shape modeling operate on triangulated meshes with precise part segmentation. Hence, they are not directly applicable to the large-scale ShapeNet dataset with less precise segmentation, preventing a fair comparison. We therefore added the following comparisons with modern neural-net-based techniques: we combined the state-of-the-art ComplementMe method [37] with a 3D-CNN segmentation network [30]. From the former we used the *component placement network*, which, given a partial shape and a complementary component, produces a 3-D translation to place the component correctly w.r.t. the partial shape. To produce the "to-be-added" component we used a 3D-CNN segmentation network, described in [30], which achieved a state-of-the-art mean Intersection over Union (mIoU) of 0.91 on the test set. Together, these two networks replace our proposed Decomposer-Composer. Both networks were trained using

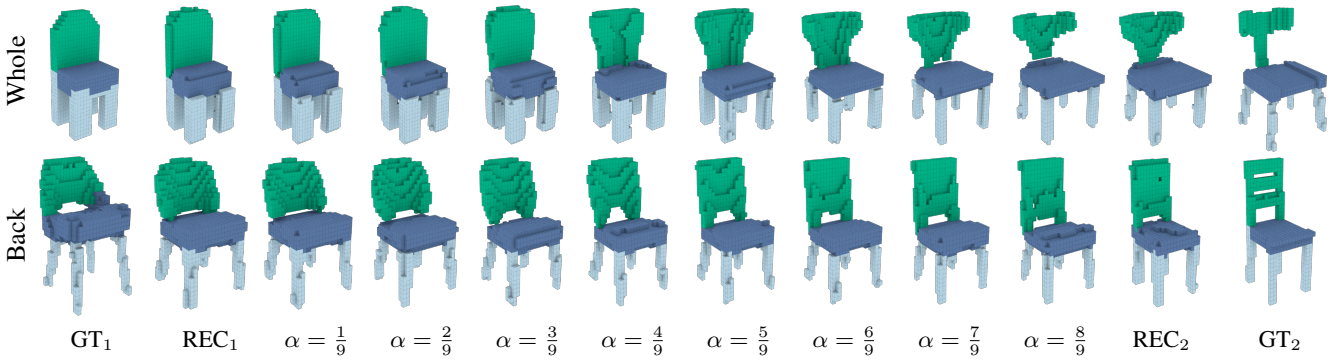

Figure 7: Example of a whole (top) and partial (bottom) shape interpolation. $GT_{1/2}$ denote original models, $REC_{1/2}$ - their reconstructions, and linear interpolation results are in the middle. Unlabeled shapes were used as an input.

Method \ Metric	mIoU	mIoU (parts)	Connectivity			Classifier accuracy			Symmetry score		
	Rec.	Rec.	Rec.	Swap	Mix	Rec.	Swap	Mix	Rec.	Swap	Mix
Our method	0.64	0.65	**0.82**	**0.71**	**0.65**	**0.95**	0.89	**0.83**	0.95	0.95	0.95
W/o cycle loss	0.63	0.66	0.74	0.62	0.54	0.93	0.84	0.80	0.96	0.96	0.95
Fixed projection	0.63	0.65	0.72	0.61	0.58	0.94	0.86	0.77	0.94	0.95	0.95
Composer w/o STN	**0.75**	**0.8**	0.69	0.48	0.23	**0.95**	**0.9**	0.71	0.95	0.91	0.85
Naive placement	-	-	-	0.68	0.62	0.61	0.47	0.21	-	**0.96**	**0.96**
ComplementMe	-	-	-	**0.71**	0.47	-	0.66	0.43	-	0.66	0.43
Segmentation+STN	-	-	-	0.41	0.64	-	0.64	0.36	-	0.77	0.77

Table 1: Ablation study results. The evaluation metrics are mean Intersection over Union (*mIoU*), per-part mean IoU (*mIoU (parts)*), shape *connectivity* measure, binary shape *classifier accuracy*, and shape *symmetry score*. Rec., Swap and Mix stand for the shape reconstruction, part exchange and random part assembly experiment results, respectively (see Section 4.2). See Section 4.4 for a detailed description of the compared methods and the evaluation metrics.

the same training data as the proposed method. This method is denoted by *ComplementMe* in Table 1.

For an additional comparison, instead of the placement network of ComplementMe we utilized the spatial transformer network. Here, the STN was trained using the ground truth shape parts, and at test time it was applied to the results of the segmentation network, described above. This method is denoted by *Segmentation+STN* in Table 1.

Finally, we compared the proposed method to a baseline shape composition network. Given ground-truth shape parts, it composes new shapes from these parts by placing them at their original locations in the source shapes they were extracted from. All the shapes in our dataset are centered and uniformly scaled to fill the unit volume, and there exist clusters of geometrically and semantically similar shapes. Thus, we can expect that even this naive approach without part transformations will produce plausible results in some cases. This method is denoted by *Naive placement* in Table 1.

See the supplementary material for an additional qualitative comparison with 3D-GAN [42] and G2LGAN [39], using $64 \times 64 \times 64$ voxelized shapes.

4.4.3 Evaluation metrics

Mean Intersection over Union (mIoU) is commonly used to evaluate the performance of segmentation algorithms [22]. Here, we use it as a metric for the reconstruction quality. We computed the mIoU for both actual-sized reconstructed parts, and scaled and centered parts (when applicable). We denote the two measures by *mIoU* and *mIoU (parts)* in Table 1.

Connectivity In part based shape synthesis, one pathological issue is that parts are often disconnected, or penetrate each other. Here, we would like to benchmark the quality of part placement, in terms of part connectivity. For each $32 \times 32 \times 32$ volume, we compute the frequency of the shape forming a single connected component, and report it as *Connectivity* in Table 1.

Classification accuracy To measure the shape composition quality of different methods, we trained a binary neural classifier to distinguish between ground-truth whole

chairs (acting as positive examples) and chairs produced by naively placing random chair parts together (acting as negative examples). To construct the negative examples, we randomly combined ground-truth shape parts, by adding a certain semantic part only once, and placing the parts at their original locations in the source shapes they were extracted from. In addition, we removed negative examples assembled from parts from geometrically and semantically similar chairs, since such part arrangement could produce plausible shapes incorrectly placed in the negative example set. The attained classification accuracy on the test set was $\sim$ 88%. For a given set of chairs, we report the average classification score. Details of the network can be found in the supplementary material. The results are reported as *Classifier accuracy* in Table 1.

Symmetry The chair shapes in the ShapeNet are predominantly bilaterally symmetric, with vertical symmetry plane. Thus, similar to [39], we evaluated the symmetry of the reconstructed shapes, and defined the *Symmetry score* as the percentage of the matched voxels (filled or empty) in the reconstructed volume and its reflection with respect to the vertical symmetry plane. We performed this evaluation using binarized reconstruction results, effectively measuring the global symmetry of the shapes. For the evaluation, we used the shapes in the test set (690 shapes), and conducted three types of experiments: shape reconstruction, single random part exchange between a pair of random shapes, shape composition by random part assembly. The experiments are described in more detail in Sections 4.1 and 4.2.

4.4.4 Evaluation result discussion

According to all metrics, our method outperforms or performs on par with all the baselines, and *significantly* outperforms other existing methods. This shows that our design choices - the cycle loss, learned projection matrices and usage of the STN, help to achieve plausible results both when reconstructing shapes, and when performing composite shape synthesis. This is especially pronounced in the connectivity test results, illustrating that these design choices are necessary for achieving good assembly quality.

In the classifier accuracy test and the symmetry test, the proposed method performs slightly better or on par with all baselines considered in the ablation study. It seems that both these tests are less sensitive to disconnected shape components, and most advantage that the proposed method achieves over the baselines is in its composition robustness. As expected, the naive placement also achieves high symmetry score, since it preserves the symmetry of the ground-truth parts during shape assembly.

According to the mIoU and per-part mIoU metrics, the proposed method performs on par with all baselines, except when using the simple version of the Composer, without STN. This follows from the fact that the proposed system, while reconstructing better fine geometry features, decomposes the problem into two inference problems, for the geometry and the transformation, and thus does not produce as faithful reconstruction of the original model as the simple decoder. Notably, this version of the architecture achieves worst connectivity scores for all compared methods, which follows from the fact that such a Decomposer is unable to faithfully reconstruct fine shape details. Please see the supplementary material for a qualitative comparison of the results of all the compared methods.

5. Conclusions and future work

We presented a Decomposer-Composer network for structure-aware 3D shape modelling. It is able to generate a factorized latent shape representation, where different semantic part embedding coordinates lie in separate linear subspaces. The subspace factorization allows us to perform shape manipulation via part embedding coordinates, exchange parts between shapes, or synthesize novel shapes by assembling a shape from random parts. Qualitative results show that the proposed system can generate high fidelity 3D shapes and meaningful part manipulations. Quantitative results shows we are competitive in the mIOU, connectivity, symmetry and classification benchmarks.

While the proposed approach makes a step toward automatic shape-from-part assembly, it has several limitations. First, while we can generate high-fidelity shapes at a relatively low resolution, memory limitations do not allow us to work with voxelized shapes of higher resolution. Memory-efficient architectures, such as OctNet [31] and PointGrid [18], may help alleviate this constraint. Alternatively, using point-based shape representations and compatible deep network architectures, such as [30], may also reduce the memory requirements and increase the output resolution.

Secondly, we made a simplifying assumption that a plausible shape can be assembled from parts using per-part affine transformations, which represent only a subset of possible transformations. While this assumption simplifies the training, it is quite restrictive in terms of the deformations we can perform. In future work, we will consider general transformations which have higher degree of freedom, such as a 3D thin plate spline or a general deformation fields. To promote better part connectivity, we will explore additional shape connectivity preservation losses, similar to [39]. Finally, we have been using a cross-entropy loss to measure the shape reconstruction quality; it would be interesting to investigate the use of a GAN-type loss in this structure-aware shape generation context.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016. 5

[2] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Representation learning and adversarial generation of 3d point clouds. *arXiv preprint arXiv:1707.02392*, 2017. 1, 2

[3] Panos Achlioptas, Judy Fan, X.D. Robert Hawkins, D. Noah Goodman, and J. Leonidas Guibas. ShapeGlot: Learning language for shape differentiation. *CoRR*, abs/1905.02925, 2019. 1

[4] Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde. Projecting embeddings for domain adaption: Joint modeling of sentiment analysis in diverse domains. *arXiv preprint arXiv:1806.04381*, 2018. 3

[5] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001. 5

[6] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 5

[7] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016. 3

[8] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 1

[9] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, volume 2, page 6, 2017. 1

[10] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016. 3

[11] Ruizhen Hu, Zihao Yan, Jingwen Zhang, Oliver van Kaick, Ariel Shamir, Hao Zhang, and Hui Huang. Predictive and generative neural networks for object functionality. In *Computer Graphics Forum (Eurographics State-of-the-art report)*, volume 37, pages 603–624, 2018. 2

[12] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 2, 4, 5

[13] Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. A probabilistic model for component-based shape synthesis. *ACM Transactions on Graphics (TOG)*, 31(4):55, 2012. 2, 4

[14] Angjoo Kanazawa, Shahar Kovalsky, Ronen Basri, and David Jacobs. Learning 3d deformation of animals from 2d images. In *Computer Graphics Forum*, volume 35, pages 365–374. Wiley Online Library, 2016. 2

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1

[17] Andrey Kurenkov, Jingwei Ji, Animesh Garg, Viraj Mehta, JunYoung Gwak, Christopher Choy, and Silvio Savarese. Deformnet: Free-form deformation network for 3d shape reconstruction from a single image. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 858–866. IEEE, 2018. 2

[18] Truc Le and Ye Duan. Pointgrid: A deep network for 3d shape understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9204–9214, 2018. 8

[19] Jun Li, Chengjie Niu, and Kai Xu. Learning part generation and assembly for structure-aware shape synthesis. *arXiv preprint arXiv:1906.06693*, 2019. 2

[20] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, 36(4):52, 2017. 1, 2, 4

[21] Chen-Hsuan Lin, Ersin Yumer, Oliver Wang, Eli Shechtman, and Simon Lucey. St-gan: Spatial transformer generative adversarial networks for image compositing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9455–9464, 2018. 2

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 7

[23] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *arXiv preprint arXiv:1908.00575*, 2019. 2

[24] Charlie Nash and Chris KI Williams. The shape variational autoencoder: A deep generative model of part-segmented 3d objects. In *Computer Graphics Forum*, volume 36, pages 1–12. Wiley Online Library, 2017. 1, 2

[25] Andy Nguyen, Mirela Ben-Chen, Katarzyna Welnicka, Yinyu Ye, and Leonidas Guibas. An optimization approach to improving collections of shape maps. In *Computer Graphics Forum*, volume 30, pages 1481–1491. Wiley Online Library, 2011. 2, 4

[26] Fakir S. Nooruddin and Greg Turk. Simplification and repair of polygonal models using volumetric techniques. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):191–205, 2003. 5

[27] Christian Poelitz. Projection based transfer learning. In *Workshops at ECML*, 2014. 3

[28] Dominic V Poerio and Steven D Brown. Dual-domain calibration transfer using orthogonal projection. *Applied spectroscopy*, 72(3):378–391, 2018. 3

[29] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 818–833, 2018. 1, 2, 3, 4

[30] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017. 6, 8

[31] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 3, 2017. 8

[32] Nadav Schor, Oren Katzir, Hao Zhang, and Daniel Cohen-Or. Learning to generate the" unseen" via part synthesis and composition. *arXiv preprint arXiv:1811.07441*, 2018. 2

[33] Lutfi Kerem Senel, Ihsan Utlu, Veysel Yucesoy, Aykut Koc, and Tolga Cukur. Semantic structure and interpretability of word embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2018. 3

[34] Chao-Hui Shen, Hongbo Fu, Kang Chen, and Shi-Min Hu. Structure recovery by part assembly. *ACM Transactions on Graphics (TOG)*, 31(6):180, 2012. 4

[35] Zhixin Shu, Mihir Sahasrabudhe, Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance, 2018. 1, 3

[36] Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. Neural face editing with intrinsic image disentangling. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5444–5453. IEEE, 2017. 1, 3

[37] Minhyuk Sung, Hao Su, Vladimir G Kim, Siddhartha Chaudhuri, and Leonidas Guibas. Complementme: weakly-supervised component suggestions for 3d modeling. *ACM Transactions on Graphics (TOG)*, 36(6):226, 2017. 6

[38] Fan Wang, Qixing Huang, and Leonidas J Guibas. Image co-segmentation via consistent functional maps. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 849–856, 2013. 2, 4

[39] Hao Wang, Nadav Schor, Ruizhen Hu, Haibin Huang, Daniel Cohen-Or, and Hui Huang. Global-to-local generative model for 3d shapes. *ACM Transactions on Graphics (Proc. SIGGRAPH ASIA)*, 37(6):214:1214:10, 2018. 1, 2, 3, 4, 7, 8

[40] Xiaolong Wang and Abhinav Gupta. Generative image modeling using style and structure adversarial networks. In *European Conference on Computer Vision*, pages 318–335. Springer, 2016. 2

[41] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016. 1, 2

[42] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016. 7

[43] Zhijie Wu, Xiang Wang, Di Lin, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Structure-aware generative network for 3d-shape modeling. *arXiv preprint arXiv:1808.03981*, 2018. 1

[44] Zhijie Wu, Xiang Wang, Di Lin, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Sagnet: Structure-aware generative network for 3d-shape modeling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2019)*, 38(4):91:1–91:14, 2019. 2

[45] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9068–9079, 2018. 1

[46] Kai Xu, Hanlin Zheng, Hao Zhang, Daniel Cohen-Or, Ligang Liu, and Yueshan Xiong. Photo-inspired model-driven 3d object modeling. In *ACM Transactions on Graphics (TOG)*, volume 30, page 80. ACM, 2011. 4

[47] Li Yi, Vladimir G Kim, Duygu Ceylan, I Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, Leonidas Guibas, et al. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (TOG)*, 35(6):210, 2016. 1, 5

[48] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017. 2, 4