Eurographics Symposium on Geometry Processing 2018
T. Ju and A. Vaxman
(Guest Editors)

Volume 37 (2018), Number 5

Learning Fuzzy Set Representations of Partial Shapes on Dual Embedding Spaces

Minhyuk Sung[1], Anastasia Dubrovina[1], Vladimir G. Kim[2], and Leonidas Guibas[1]

[1]Stanford University
[2]Adobe Research

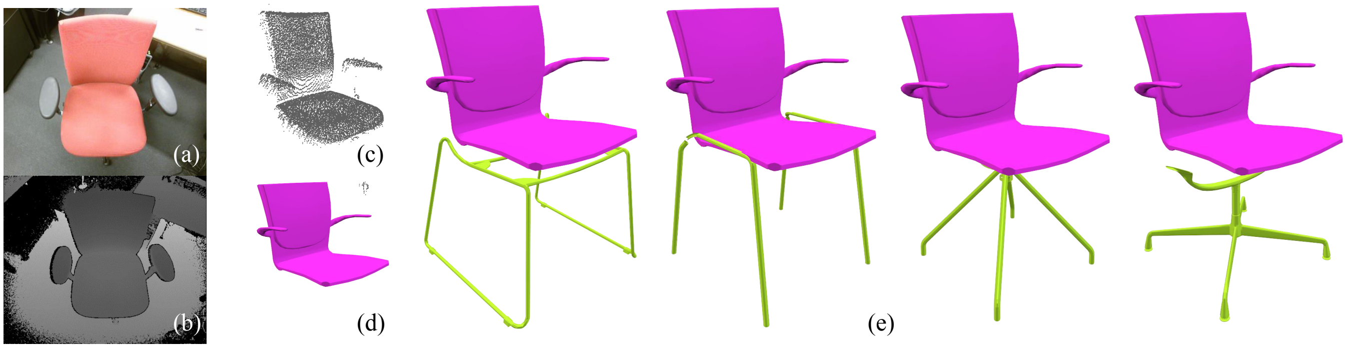


Figure 1: *An example of completing a partial scan data with various complement suggestions using our method. Partial objects are represented as fuzzy sets in our embedding, and their complements can be retrieved with a fuzzy set operation in the embedding space. (a) and (b) are a Kinect color and depth image from [SKAG15], (c) is a point cloud after removing background, (d) is a partial shape retrieved from our dataset using ICP with a manual initial pose, and (e) are complete objects with four out of the best ten complement retrievals (green parts). The position of complements are automatically computed using a placement network in [SSK*17], which is retrained with the partial object data.*

Abstract

Modeling relations between components of 3D objects is essential for many geometry editing tasks. Existing techniques commonly rely on labeled components, which requires substantial annotation effort and limits components to a dictionary of predefined semantic parts. We propose a novel framework based on neural networks that analyzes an uncurated collection of 3D models from the same category and learns two important types of semantic relations among full and partial shapes: complementarity and interchangeability. The former helps to identify which two partial shapes make a complete plausible object, and the latter indicates that interchanging two partial shapes from different objects preserves the object plausibility. Our key idea is to jointly encode both relations by embedding partial shapes as fuzzy sets in dual embedding spaces. We model these two relations as fuzzy set operations performed across the dual embedding spaces, and within each space, respectively. We demonstrate the utility of our method for various retrieval tasks that are commonly needed in geometric modeling interfaces.

CCS Concepts
• ***Computing methodologies*** → ***Shape modeling; Machine learning approaches;*** *Shape analysis;*

1. Introduction

Component-based 3D object modeling is common in design of man-made objects. Creating automatic or semi-automatic tools for such component-based modeling has been a long-standing goal in 3D object processing. Towards that goal, previous work leveraged 3D CAD model datasets with known components and component structures for creating new objects [FKS*04], completing partial shapes [SFCH12], and for analyzing shape structures [KLM*13, LXC*17]. Recently created large-scale 3D CAD datasets [Tri17, CFG*15] offer significant diversity of component geometry and structure, thus increasing the potential of component-based geometry processing for practical usages.

While a large database increases the diversity in modeling and editing shapes, it also requires the burden of annotating shape components in a consistent manner. Most previous part labeling methods are not scalable [vKXZ*13], or require some human supervision [YKC*16]. To avoid the labor of an annotation, while still taking advantage of the large scale of the databases, Sung *et al.* [SSK*17] proposed an annotation-free component assembly method, which trains a neural network to retrieve plausible complementary *components* given a query partial object. The relations among partial objects and complementary components are learned from the data. However, the incremental approach of Sung *et al.* does not account for the plausibility of the full constructed shape, or can detect *groups of segments* which can plausibly complete a given partial shape. This functionality is important when the plausibility of the full object is important, for example when creating a new object by mixing components from different models in the database [XZCOC12], or when completing partially created objects [SKAG15].

To address these limitations, we propose an annotation-free deep learning framework which learns partial shape representations from database component assemblies, and jointly encodes two semantic relations between partial shapes: *complementarity* and *interchangeability*. Complementarity means that the two partial shapes can be combined into a complete, semantically meaningful object. Interchangeability indicates that replacing a part of a model with another partial shape still produces a plausible new object. This relations can capture semantic similarities among partial shapes in terms of their usage in the context of full objects, even when these partial shapes are geometrically dissimilar. Both complementarity and interchangeability are closely related to each other, since interchangeability means that two partial shapes share the same set of complements.

Encoding these relations in embedding spaces is not trivial. Complementarity is an *irreflexive* relation, therefore a naïve embedding scheme which minimizes distances among related data is not applicable. In addition, we do not have any supervision for learning interchangeability relations, and need to infer this from the complementarity relations between partial shapes. To tackle these challenges, we suggest a novel embedding approach into *dual* embedding spaces (Figure 2). We consider the symmetric (undirected) complementary relation as both-way directed relations, and create two embedding spaces f and g for one-to-N mapping, such that all variations of partial shapes are present in both spaces. Given two partial shapes, complementarity between them is reflected by their embeddings into the two spaces. To learn the N-to-N irreflexive complementarity mapping with this embedding scheme, we use fuzzy set representations [Zad65] for both embedding spaces, and encode the complementarity relation as the intersection of sets. When learning the complementary relations *across* two embedding spaces, the similarity in the *same* embedding space can be interpreted as interchangeability.

Key contributions:

- We propose a novel dual embedding framework to learn complementarity and interchangeability relations between partial shapes.

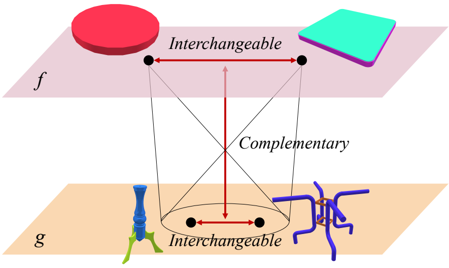


Figure 2: *We propose a dual space embedding that encode both complementarity and interchangeability. The complementarity is represented by inter-space relations, and the interchangeability is represented by intra-space relations.*

- The complementarity and interchangeability of partial shapes are encoded as inter- and intra-relations in the dual embedding spaces, respectively.
- Fuzzy set representations are utilized for both embedding spaces, to learn N-to-N irreflexive complementarity mapping between them.
- We demonstrate the effectiveness of the proposed embedding scheme for learning the two relations between partial shapes for several shape modeling tasks, on a variety of shape categories.

2. Related Work

We review related work on component-based 3D modeling, and structural embedding techniques using neural networks.

Component-based 3D Modeling Funkhouser *et al.* [FKS*04] were the first to introduce the idea of reusing parts in the existing 3D models for creating new objects. Subsequent approaches [CKGK11, KCKK12, CKGF13] developed the idea of shape construction using labeled components, by learning the component structure and suggesting appropriate components in the interactive assembly process. Shen *et al.* [SFCH12] used partial scanned data as a cue for constructing complete 3D models. There, components in the database were retrieved and stitched to each other to fit to the input geometry and fill the missing area in the incomplete scans. This completion approach was further extended by Sung *et al.* [SKAG15], who integrated both symmetry- and retrieval-based inferences to detect the missing parts. These approaches successfully demonstrated practical applications of leveraging the component structure in a given dataset of 3D models, but all of them relied on having models with labeled components, which required significant annotation effort.

Recently, Sung *et al.* [SSK*17] introduced a method for constructing shapes from *unlabeled* components in an iterative assembly process. Specifically, given a partial shape, their method produces multiple plausible complementary components. This is done by training a neural network which jointly maps database components into the embedding space and predicts a complementary component probability distribution over that space given a partial shape. While here we also learn complementary relations, we aim at learning relations between *partial shapes*, as opposed to relations between partial shapes and single components in [SSK*17]. Towards that goal, we embed all possible partial objects and discover shapes that complete the query in a plausible manner. This is more challenging problem since the shape variation space of partial shapes is much larger than the space of single components. We evaluate both methods using partial shape datasets, and demonstrate that our method outperforms the previous work in quantitative and qualitative evaluations.

Deep Structural Embedding Data embedding with neural networks is widely used for encoding relations among large-scale data. The advantages of inferring relations between data points in a structured embedding space, instead of learning binary indicator functions of the relations, are clearly described in the seminal work [BWCB11]: simple adaptation to various datasets, compact information storage, flexible joint encoding of different types of relations, and, most importantly, the ability to infer unseen relations from the structure of the embedding space. Notable examples of the above in language processing are [SCMN13] and [MSC*13], where low-dimensional word embeddings were used to capture the relations among words, and in image processing - [SKP15], which proposed learning an embedding space of facial images, where distances directly corresponded to face similarity, independently of the face pose. Deep embedding of 3D data was utilized by Li *et al*. [LSQ*15], who suggested a method for real time object reconstruction, by learning correlation between images and 3D models, and by Wu *et al*. [WZX*16], who showed that the relations among 3D object structures can be learned using a generative adversarial network. Sung *et al*. [SSK*17] learned a different embedding space of complementary components for input partial shapes.

Unlike previous work, in the proposed approach we construct embedding spaces that reflect both complementarity relations, learned in a supervised manner, as well as interchangeability relations, for which no supervision is provided, and successfully discover both types of relations between previously unseen partial shapes. In addition, the majority of existing techniques for deep embedding encode reflexive and symmetric relations, such as similarities, into distances or angles between vectors in the embedding space [KFF15]. Some recent methods focus on other types of relations, such partial order relations [VKFU16, NK17], which are asymmetric and transitive. In this work, we consider different *irreflexive* and symmetric complementarity relations, and propose a new embedding space construction technique to reflect these relations.

3. Method

In this section, we explain how we train a neural network to jointly encode both the complementarity and interchangeability in embedding spaces. We first give an overview of the proposed approach, then describe in detail all its components.

3.1. Overview

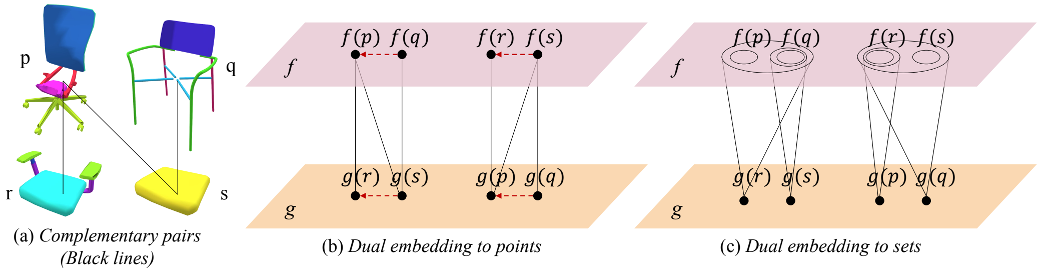
We design binary energy functions of complementarity and interchangeability, both of which take the embedding coordinates of partial shapes as inputs. A neural network is to used to define the embedding function for partial shapes. In the training of the neural network, we create complementary pairs of partial shapes as training examples by splitting full objects into two parts. But we do not have any supervision for interchangeability. Thus, the network is trained to minimize only the complementarity energy, but is still able to predict the interchangeability from the embedding structure. In next subsections, we elaborate on how we define the embedding spaces and the binary energy functions on them. We first describe the motivation for using the dual embedding spaces to represent complementarity (Section 3.2), and how this relation is encoded across the dual spaces (Section 3.3). Then, we define the complementarity and interchangeability energy functions as fuzzy set operations on the embedding spaces, in Sections 3.4 and 3.5, respectively. The loss functions used for the neural network training and the neural network implementation details are described in Sections 3.6 and 3.7.

3.2. Dual Embedding Spaces

We first describe how to design the embedding spaces and a binary indicator function for complementarity. One can consider a graph which nodes and edges indicate partial shapes and their complementary relations, respectively. Our problem can be viewed as a graph embedding problem aimed at finding unseen edges between nodes (unseen complementarity relations) by using the geometry of partial shapes as node attributes. The binary complementarity indicator function is then defined as whether a partial shape is connected to the other, and vice versa. There are several previous techniques for the graph node embedding problem with input node attributes [PARS14, HYL17, KW17], but they all use same approach of mapping neighboring nodes to proximal locations in the embedding space, which is not applicable to our case for two reasons. First, complementarity is *not* transitive, meaning that, given a partial shape, a complement of its complement is generally not a complement of it. Thus, first-order neighbors need to be discriminated from higher-order neighbors. Second, complementarity is also *not* reflective; a partial shape is not a complement of itself. Thus, each node should be isolated from the its neighbors in the embedding space. To handle the *non-transitivity* and *irreflexivity*, we consider *dual* embedding spaces as shown in Figure 2. All partial shapes live in both embedding spaces, and the complementarity relations among partial shapes are now represented by the relations between their embedding coordinates *across* the two spaces. Since one partial shape can have different embedding coordinates in different spaces, this allows a partial shape not to have a relation to itself. We define the structure of two embedding spaces and the inter-space complementary relation in the next subsection.

3.3. Embedding as Set Inclusion

The naïve idea for constructing dual spaces is to align the positions of complementary partial shapes at the same coordinates in different spaces. However, it may cause some non-complementary pairs to be encoded as complementary to each other. For example, in the case illustrated in Figure 3(a), shape p is complementary to both r and s, but shape q is only complementary to s. Then, from the complementarity of p and r, s, both r and s should be placed at the location of p in the other space ($f(p) = g(r) = g(s)$ and $f(r) = f(s) = g(p)$). Because of the complementarity of q and s,



(a) *Complementary pairs (Black lines)* (b) *Dual embedding to points* (c) *Dual embedding to sets*

Figure 3: *A schematic example showing how our set representation works on the dual embedding space. (a) (p, r) (p, s), and (q, s) are complementary pairs, and (q, r) is a non-complementary pair. (b) A case of embedding data as points, and aligning embedding coordinates of complementary shapes in different spaces (e.g. f(p) = g(s) and f(r) = g(p)). The complementarity of (p,r) and (p,s) implies that $g(r) = g(s)$ (similarly, $f(r) = f(s)$). Because of the complementarity of (q,s) this also* incorrectly *implies that $g(q) = g(r)$ ($f(q) = f(r)$), which results in a wrong complementary relation between q and r. (c) The proposed approach of representing complementarity with* set representations. *All relations can be correctly encoded.*

q also goes to s's location in the other space. This leads to a complementary relation between q and r, which is contradictory to the assumption (See Figure 3(b)).

To avoid this problem, we suggest to relate a coordinate in one embedding space to *multiple* coordinates in the other space. In the aspect of graph embedding described in Section 3.2, the relation is indicated by checking both ways of whether a node is a neighbor of the other node. When considering one-way relations only, and encoding 1-to-N mapping from a partial shape to its complements, we view the embedding coordinates as a representation of a *set*, such that the complementarity relation is encoded as *inclusion* from one space to the other. One choice of encoding sets and inclusions in the embedding space is using the approach of Vendrov *et al*. [VKFU16]:

$$x \subseteq y \quad \Leftrightarrow \quad \bigwedge_{i=1}^{D} x_i \leq y_i, \tag{1}$$

where $x, y \in \mathbb{R}_+^N$ are the embedding coordinates of x, y, respectively, and $\bigwedge$ is a 'logical and' operator (note that Vendrov *et al.* use reversed direction of the inequality but we switch back to the natural direction). Since here we wish to relate embedding coordinates in two different spaces (due to the irreflexivity), given two embedding spaces $f(\cdot)$ and $g(\cdot)$ we represent the *one-way* complementarity $c(x \to y)$ as follows:

$$c(x \to y) \quad \Leftrightarrow \quad \bigwedge_{i=1}^{D} f(x)_i \leq g(y)_i. \tag{2}$$

According to the analogy with the inclusion relation, we will call f and g the *subset* and *superset* spaces, respectively, in the rest of paper. Then, the binary indicator function for *both-way* complementarity $c(x,y)$ can be represented as follows:

$$c(x,y) \quad \Leftrightarrow \quad c(x \to y) \wedge c(y \to x). \tag{3}$$

Figure 3(c) illustrates how the relations between p,q,r,s in Figure 3(a) can be represented with sets and inclusions.

3.4. Fuzzy Set Interpretation

While the inclusion embedding of [VKFU16] was first applied for neural network training in that paper, the idea is actually closely related to the well-studied *fuzzy set theory* [Zad65]. In fuzzy set theory, a set is represented with fuzzy memberships over a discrete set of elements, which is also called *possibility* distribution. Then, the notion of inclusion is defined so that the membership scores (or probabilities) over all elements of a superset are greater or equal to the corresponding scores of a subset, which is identical with the definition in Equation 1.

With the fuzzy set representation, one can consider how to define set operations analogous to classic crisp (non-fuzzy) set theory. For example, for 'logical and' (intersection) $\wedge$, and 'inclusive or' (union) $\vee$, there are various ways of defining the operations with fuzzy sets (see Section 3 in [Zim01] for details), but the simplest form is using minimum and maximum operations:

$$\begin{aligned}(x \wedge y)_i &= \min(x_i, y_i),\\ (x \vee y)_i &= \max(x_i, y_i).\end{aligned} \tag{4}$$

In the neural net training, we need to *fuzzify* the notion of inclusion to obtain a continuous loss function. Vendrov *et al.* suggest to penalize when the embedding coordinates of a subset are greater than the coordinates of the superset, element-wise:

$$E(x \subseteq y) = \sum_i \max(0, x_i - y_i)^2. \tag{5}$$

This is actually the same as making the subset to be equal to the intersection of the two sets x and y, using the above definition of intersection:

$$E(x \subseteq y) = \sum_i (x_i - \min(x_i, y_i))^2 = \|x - (x \wedge y)\|_2^2. \tag{6}$$

Using Equation 2 and Equation 3, we can define the energy functions for the one-way complementarity $E_c(x \to y)$ and the both-way complementarity $E_c(x,y)$, as follows:

$$E_c(x \to y) = \sum_i \max(0, f(x)_i - g(y)_i)^2, \tag{7}$$

$$E_c(x,y) = E_c(x \rightarrow y) + E_c(y \rightarrow x). \tag{8}$$

3.5. Interchangeability via Complementarity

In Section 3.3 and 3.4, we described how complementarity is represented as an *inter*-space fuzzy set operation. Now we discuss how we define an *intra*-space fuzzy set operation that measures the degree of *interchangeability*.

It is obvious that two partial shapes have exactly same complements with same energy values when they have the same embedding coordinates in both embedding spaces. This implies that two partial shapes with similar embedding coordinates in each embedding space have a similar set of complements. In the following propositions, we show that how the *union* and *intersection* of two fuzzy sets represented by the embedding coordinates are related to the complementarity energies with arbitrary partial shapes.

Proposition 1 $\max\left(E_c(x \rightarrow z), E_c(y \rightarrow z)\right) \leq E_c((x \vee y) \rightarrow z)$

Proof Refer to Appendix □

Proposition 2 $\max\left(E_c(z \rightarrow x), E_c(z \rightarrow y)\right) \leq E_c(z \rightarrow (x \wedge y))$

Proof Refer to Appendix □

Corollary 1

$$\begin{aligned}&\frac{1}{2}\left(E_c(x,z) + E_c(y,z)\right)\\ &\leq \max\left(E_c(x \rightarrow z), E_c(y \rightarrow z)\right) + \max\left(E_c(z \rightarrow x), E_c(z \rightarrow y)\right)\\ &\leq E_c((x \vee y) \rightarrow z) + E_c(z \rightarrow (x \wedge y))\end{aligned}$$

Corollary 1 shows that the union on the *subset* space and the intersection on the *superset* space bound the sum of complementarity energies for arbitrary partial shapes. When restrict the $l2$-norm of all embedding coordinates to be one, one can consider how close the $l2$-norm of $f(x) \vee f(y)$ and $g(x) \wedge g(y)$ are to one as a measure of *interchangeability* energy $E_r(x,y)$:

$$\begin{aligned}E_r(x,y) &\triangleq (\|f(x) \vee f(y)\|_2^2 - 1) + (1 - \|g(x) \wedge g(y)\|_2^2)\\ &= \|f(x) \vee f(y)\|_2^2 - \|g(x) \wedge g(y)\|_2^2\end{aligned} \tag{9}$$

Note that the unit $l2$-norm constraint is not only for defining interchangeability relation in the embedding, but it is also a common constraint in neural network training to avoid over-fitting [SKP15, VKFU16].

3.6. Neural Network Loss

The loss function for the neural network training is defined using the complementarity energy (Equation 8). Given N complementary pairs (x_i, y_i) in a batch, we consider all mis-matched pairs $(x_i, y_{j \neq i})$ as negative examples. We suggest two loss functions that can be used depending on the application. For complement retrieval tasks, we use pairwise ranking loss as introduced in [KFF15, VKFU16]:

$$\begin{aligned}L_R = &\sum_i \sum_{j \neq i} \max\left(0, E_c(x_i, y_i) - E_c(x_i, y_j) + \alpha\right)\\ &+ \sum_j \sum_{i \neq j} \max\left(0, E_c(x_i, y_i) - E_c(x_i, y_j) + \alpha\right),\end{aligned} \tag{10}$$

where α is a given margin parameter. We use $\alpha = 0.05$ in all our experiments. This pairwise ranking loss learns *relative* distances between positive and negative pairs. But we need a commensurable measure of complementarity for the interchangeability measure (Equation 9) since it is based on the upper bound of the complementarity energies. Thus, we introduce another loss function learning absolute errors with a threshold:

$$\begin{aligned}L_T = &\frac{1}{N} \sum_i \max\left(0, E_c(x_i, y_i) - \left(t - \frac{1}{2}\alpha\right)\right)\\ &+ \frac{1}{N(N-1)} \sum_i \sum_{j \neq i} \max\left(0, \left(t + \frac{1}{2}\alpha\right) - E_c(x_i, y_j)\right),\end{aligned} \tag{11}$$

where t is a learnable threshold parameter. This loss function makes the energy of positive pairs smaller than t and the energy of negative pair greater than t with the margin α between them.

3.7. Neural Network Architecture and Training

We used the PointNet architecture [QSMG17] as a basic building block of the proposed embedding network, as illustrated in Figure 4. Since all partial shapes are embedded into both embedding spaces, we have two separate PointNet Siamese architectures for learning the f and g embedding functions. We feed both f and g-networks with $1k$ sampled points x and y, to produce $f(x), g(x)$ and $f(y), g(y)$. Note that we use the unit $l2$-norm constraint for output embedding coordinates as described in Section 3.5. Both networks receive inputs as *centered* point clouds of partials shapes, translated so that the centers of axis-aligned bounding boxes are located at the origin. Thus, the relation prediction is performed without the information of the partial shape location. In all experiments, we train the network for 1,000 epochs with batch size of 32, and use ADAM optimizer with 1.0E-3 initial learning rate, 0.7 decay rate and 200K decay steps. The embedding dimension is fixed to be $D = 100$.

4. Results

4.1. Partial Shape Dataset

In our experiments, we use the ShapeNet 3D component dataset of Sung *et al.* [SSK*17]. It consists of 9 model categories from the ShapeNet [CFG*15], each of which has up to 2.4K models. All models are consistently aligned, scaled to have unit radius, and pre-segmented into components. The components were created from the ShapeNet CAD model scene graphs; the leaf node components of scene graphs were preprocessed, so that symmetric components were grouped into a single component, and small components were merged with the adjacent larger components. We build *contact* graphs of components based on their proximity, and during training generate complementary partial shape pairs by randomly splitting the contact graphs into two subgraphs. The dataset

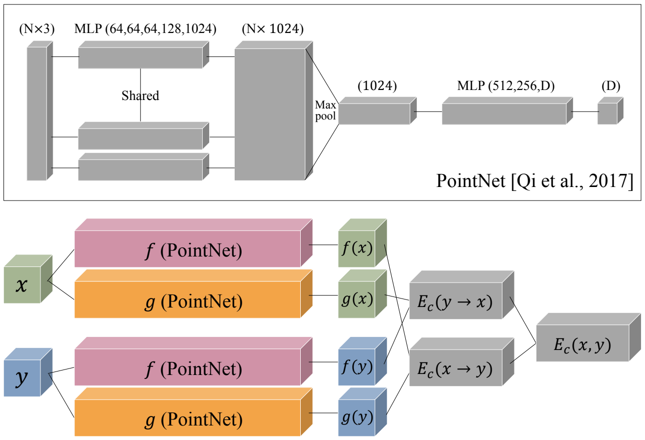


Figure 4: *Neural network architecture. We build two Siamese networks with shared weights, for both f and g embeddings. We use PointNet [QSMG17] to compute the embeddings, but any neural network architecture for 3D geometry processing is applicable.*

is split into 80% training and 20% test sets, and separate networks are learned for different model categories.

4.2. Qualitative Evaluation

Complementarity Evaluation Figure 5 shows examples of the top-5 complement retrieval, in terms of the complementarity energy E_c (Equation 8), one example per category. In the retrieval experiment, we used all possible partial shapes in the test set as a database of complement candidates. The centered query and retrieved partial shapes were automatically stitched, using the placement neural network introduced in [SSK*17]. Here, the placement net was retrained with partial shapes instead of single components. We note that the retrieved complementary partial shapes have different geometries and styles, but most of them complement the queries in a plausible manner. For example, given a chair query with swivel legs removed, both four legs and swivel legs are retrieved, and all the results look plausible. In another example, given the stretcher of a table, different tables having appropriate widths fitting the stretcher are retrieved. The three lamps are complemented with suitable mount accessories, including the three wall mount plates of the last complement.

Interchangeability Evaluation Figure 9 shows examples of interchangeable partial shapes extraction. Here, we also used all possible partial shapes in the test set as a database of interchangeable candidates. For each query shape, we extracted its top-5 nearest partial shape neighbors, now using the interchangeability measure E_r (Equation 9). The results demonstrate that our method can properly learn interchangeability among partial shapes, even when they are constructed of different components and have dissimilar geometries. For instance, table stand bottoms (Figure 9, second column from the right) have different shapes, but they can be replaced by each other in any table. For the partial chair without a seat, we successfully retrieve all partial chairs without seats, while all retrievals have different back and leg parts. The lamp components retrieved by a query have various shapes with different sizes, but all of them are shade parts with tubings.

4.3. Comparisons

We compare our method with ComplementMe [SSK*17], which also learns complementary relations among 3D shapes, but for a different purpose. ComplementMe was designed for an iterative component assembly task, therefore it retrieves a single *component* for each query partial shape in every iteration. Thus, when applied to fully automatic shape synthesis or completion, ComplementMe has limitations of accumulating noise in successive iterations and missing a notion of termination. Contrarily, the proposed method finds *groups of components* fully completing the query shape in a single retrieval step. Another difference in terms of the difficulty of the problems is that the proposed method handles much larger shape variation space since it embeds all possible partial shapes, while ComplementMe only embeds single components.

ComplementMe approach can also be adapted to handle partial shapes instead of individual components. However, we argue that the proposed method is more effective for learning both complementarity and interchangeability relations due to the differences in relation representations. In ComplementMe, the one-way complementarity energy function is defined as a negative log-likelihood of a Gaussian mixture probability density function. The multi-modality of the distribution is essential in ComplementMe since a single Gaussian raises the embedding collapse problem described in Section 3.2 and Figure 3(b). But it also leads to two limitations. First, a larger number of Gaussians can better encode all possible complementary relations, but it also increases the number of output parameters, making the network more difficult to train. Thus, the representation power can be impaired either when the number of Gaussians is too small or too large. Second, some interchangeability relations may not be captured with the multi-modal distributions since two interchangeable partial shapes can be included in different modes. Our fuzzy-set-based representation, using a single vector to represent a partial shape, is much more concise than the Gaussian mixture representation and does not have the above multi-modality issues.

In the following experiments, we demonstrate that the difference in the relation representations affects the performance of both complementary and interchangeable shape retrievals in practice. For comparison, we re-train the ComplementMe retrieval network using our partial shape dataset, and the same training parameters and embedding dimension as in the proposed method. We also use our loss functions (Equation 10 and Equation 11) instead of the triplet loss in ComplementMe. Note that the losses are identical except for the larger number of negative pairs used in the proposed loss ($N(N-1)$ vs. N in ComplementMe, which makes the training more efficient).

Evaluating whether the retrieved shapes are complementary or interchangeable is non-trivial since the criteria are subjective. Human annotations may not be consistent and can be prone to bias. Thus, for the quantitative evaluation of the complement retrievals, we measure recall and rank of the *ground truth* complements, following the recent retrieval work [KLSW15, VKFU16]. Table 1

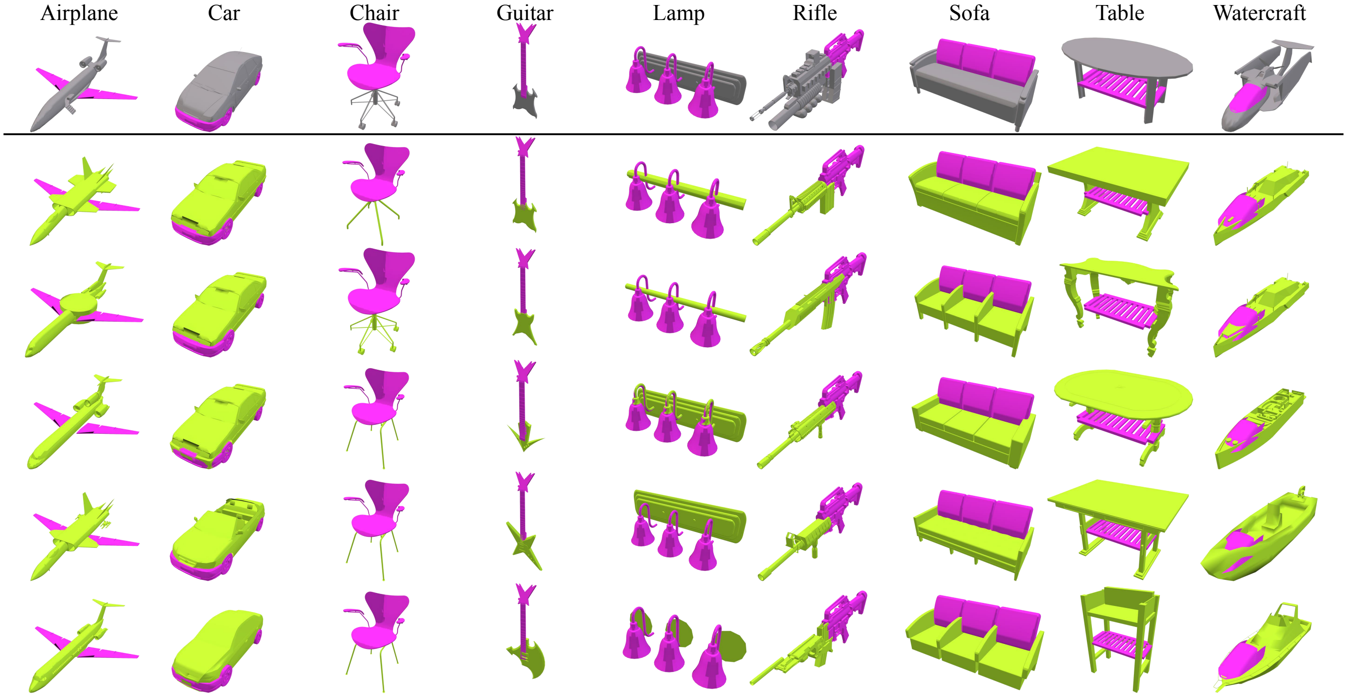


Figure 5: *Examples of top-5 complement shape retrievals. The top row shows the original shapes (gray) with highlighted query partial shapes (magenta). Next five rows show the retrieved top-5 complementary partial shapes (green), together with the query shapes (magenta). See the accompanying text for details.*

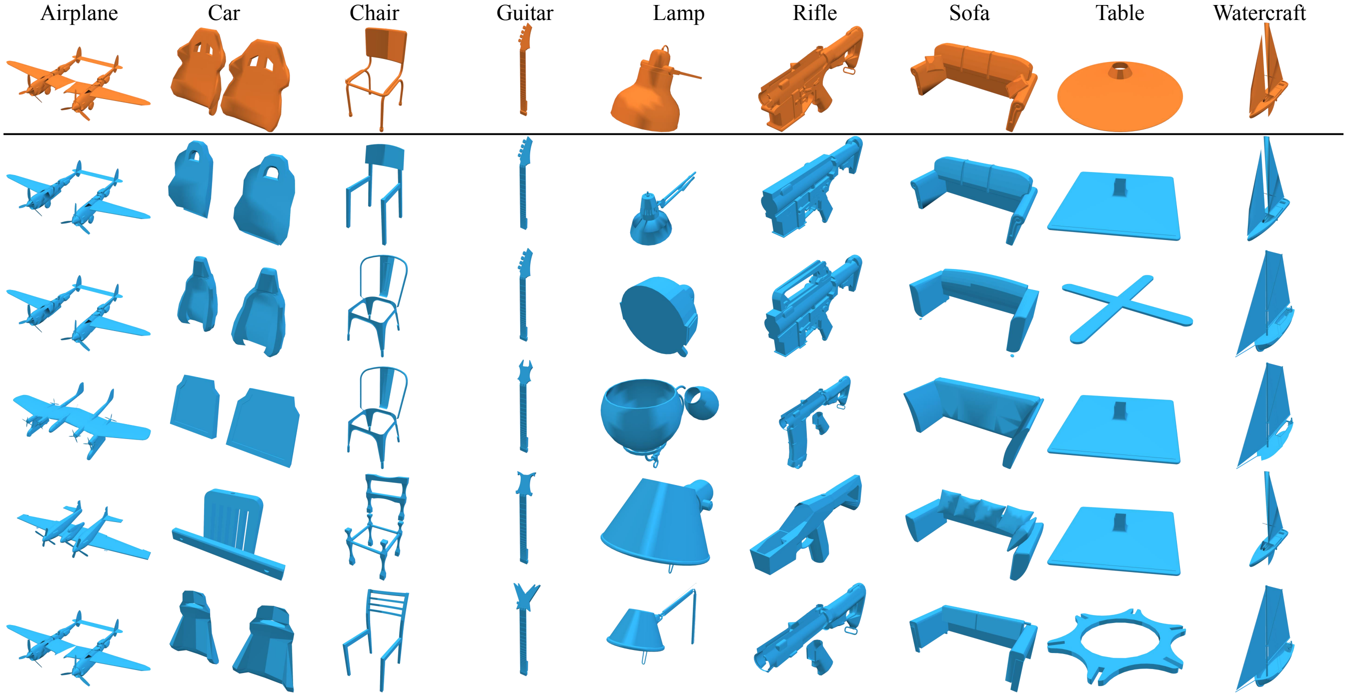


Figure 6: *Examples of partial shape interchangeability. The top row shows the query partial shapes. Next five rows show partial shapes interchangeable with it, which were detected using the proposed approach. Different components comprising the partial shapes are shown with different shades of blue. See the accompanying text for details.*

Category (# Partial Shapes)		Airplane (4140)	Car (5770)	Chair (8374)	Guitar (198)	Lamp (1778)	Rifle (1184)	Sofa (4452)	Table (4594)	Watercraft (1028)	Mean
Recall@1	CM	9.9	2.4	4.9	19.2	1.7	1.9	3.9	2.7	0.7	4.3
	Ours	**17.5**	**5.8**	**8.0**	**23.7**	**5.1**	**7.3**	**6.7**	**4.1**	**3.2**	**7.8**
Recall@10	CM	48.6	15.5	27.2	67.7	11.1	17.1	20.0	15.5	7.3	23.5
	Ours	**61.3**	**30.5**	**35.0**	**72.2**	**19.7**	**23.5**	**30.1**	**19.2**	**14.3**	**32.9**
Median Percentile Rank	CM	99.8	98.8	99.6	97.0	89.6	94.3	98.5	98.3	87.0	97.9
	Ours	**99.9**	**99.5**	**99.7**	**98.5**	**90.4**	**95.8**	**99.2**	**98.5**	**88.7**	**98.4**
Mean Percentile Rank	CM	98.4	96.4	98.3	**94.5**	**81.4**	88.2	94.0	94.9	77.6	94.8
	Ours	**98.5**	**97.2**	**98.5**	93.8	79.9	**89.0**	**94.9**	**95.0**	**78.7**	**95.2**

Table 1: *Quantitative evaluations of complement retrievals using ComplementMe [SSK*17] and our method. The number of all possible partial shapes in the test set are given in parentheses at the first row. Recall@N measures the percentage of the ground truth complements in the top-N rank retrievals, and Percentile Rank measures the percentage of partial shapes having ranks equal or greater than the rank of the ground truth complements. Higher is better in all measures.*

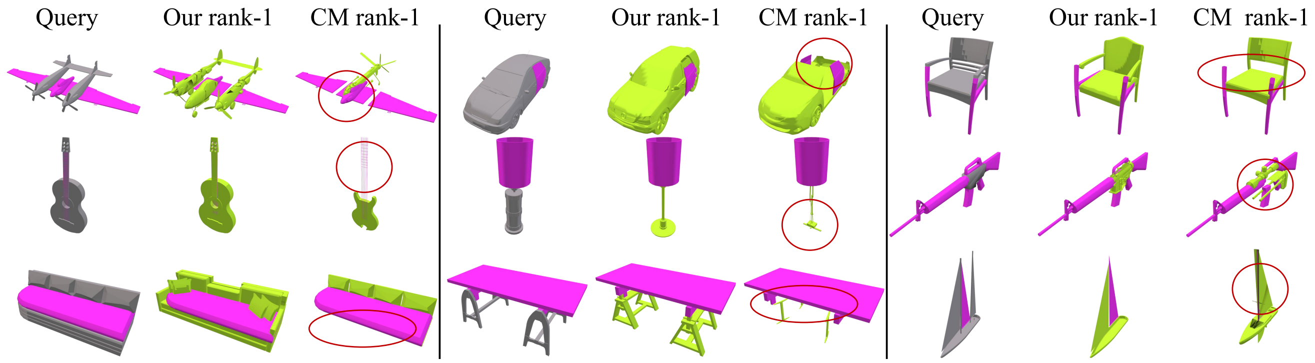


Figure 7: *Comparison of complement shape retrieval results of the proposed method and the ComplementMe architecture. In each case, the first column is the query partial shape, the second column is our rank-1 retrieval, and the third column is the rank-1 retrieval of ComplementMe [SSK*17]. The color-coding is the same as in Figure 5. The red circles show incomplete areas in the ComplementMe results.*

shows the quantitative evaluation results when testing both ComplementMe and our method with all possible partial shapes in the test set as queries (the partial shape number are in parentheses at the first row, for each shape category). Recall@N indicates the percentage of the ground truth in the top rank-N retrievals, and percentile rank indicates the percentage of partial shapes having ranks equal or greater than the rank of the ground truth. The proposed fuzzy set representation outperforms ComplementMe in all cases, except for mean percentile ranks for two model categories.

Figure 7 shows complementary shape retrieval results of both methods. Although ComplementMe produces reasonable results, some retrievals do not fully complement the query shape, resulting in missing areas in the combined shapes: e.g., armrests in a chair, a trunk in a car, and the bottom parts of a sofa and a lamp. Our method successfully creates complete plausible output shapes with the retrieved complements.

For the interchangeable shape retrievals, we also compare the proposed method with ComplementMe, and additionally with Multi-View CNN (MVCNN) descriptor [SMKLM15]. To evaluate the retrievals quantitatively, we use *semantic* single parts [YKC*16] instead of partial shapes, and measure the correlation between the detected interchangeable parts and their semantic labels. While this measure is imperfect since some parts with different labels may have similar shapes, this is uncommon for most shape categories. Given a query semantic part, we find its interchangeable parts using each one of the methods: the interchangeability measure $E_r(x,y)$ (Equation 9) for our method, and the distances in the embedding and descriptor spaces for ComplementMe and MVCNN, respectively. Then, we compute the ratio of the number of neighbors having the same semantic label as the query and the number of all retrieved neighbors. Higher values mean that more neighboring parts share the semantic label with the query. Figure 8 shows the average ratio of equal nearest neighbor labels, as a function of the number of the neighbors, up to 10% of the all parts. Our method (green line) consistently produces higher correlation with semantic labels compared to other methods in all categories, except cars, for which training data mostly have coarse segmentations not into parts but into larger partial shapes.

Figure 9 visualizes some results of interchangeable partial shape retrievals using both methods: query shape, rank-1 retrieval of ours, and rank-1 retrieval of ComplementMe, from left to right. We also list the retrieval ranks based on the interchangeability measure (embedding distances for ComplementMe and energy function $E_r(x,y)$ for ours) of the other method. The results indicate that the proposed

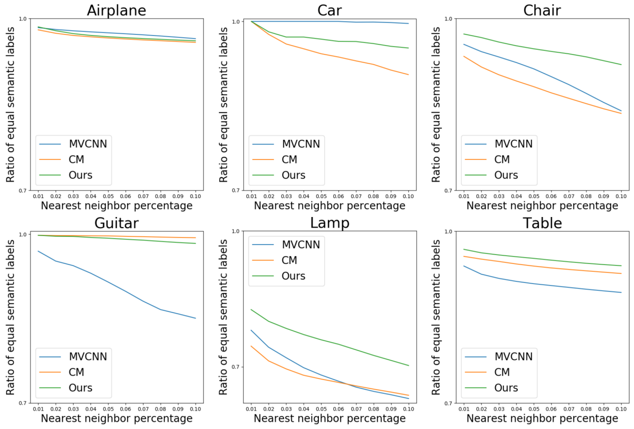


Figure 8: *Evaluation of interchangeability measures with human-annotated part correspondences in [YKC*16]. We present correlations between the interchangeability measures and the semantic part labels. Our method (green) shows better performance, indicated by higher correlations, as compared to MVCNN (blue) [SMKLM15] and ComplementMe (orange) [SSK*17]. See the accompanying text for details.*

method is able to detect more semantically meaningful shapes as its rank-1 retrievals, as compared to ComplementMe. Furthermore, according to the ComplementMe ranks, it maps both the shapes incorrectly retrieved by it and the plausible shapes retrieved by the proposed method nearby in its embedding space. In contrast, the proposed method is able to discriminate between the interchangeable and non-interchangeable shapes, as indicated by its ranks for the shapes retrieved by ComplementMe.

4.4. Application - Partial scan completion

One potential application of the proposed method is completion of partial scan data with various partial shapes form the dataset. Figures 1 and 10 show examples of completing real and synthetic partial scan data with complements retrieved with our method. We first use ICP with manual initial scan pose, to find a partial shape in our test dataset that best fits the input point cloud (shown with pink). We then retrieve complementary partial shapes (shown with green) using our fuzzy set operations. Note that, unlike other shape completion methods, such as [SKAG15, DQN17], which create a single completion result, our method can easily provide multiple plausible outputs according to the partiality of the input data.

4.5. Computation time

Both training and test are performed with a single NVIDIA TITAN Xp graph card. It took 4.5 hours to train a network for 50*k* iterations. At test time, computing the embedding coordinates of 5*k* partial shapes and their corresponding complementarity/interchangeability energies w.r.t. the query, takes in a few seconds.

5. Conclusion and future work

We have presented a novel neural network-based framework for learning complementarity and interchangeability relations between partial shapes. The two relations are learned jointly by embedding the partial shapes into dual embedding spaces, where the shapes are encoded using the fuzzy set representation. This embedding allows us to model the complementarity and the interchangeability as fuzzy set operations performed across and within the embedding spaces, respectively. The method is fully automatic, and was trained using a dataset of models with unlabeled components. Qualitative and quantitative evaluations demonstrate that our method captures well both types of relations, and produces meaningful results when applied to previously unseen shapes.

While our framework is applicable to partial shape completion, it is limited to filling the missing area at the level of the components, and does not facilitate symmetry information as done in [SKAG15] (Figure 11(a)). Also, small or thin components can be neglected in the retrieved shapes due to the limited resolution of point clouds used as neural network inputs (Figure 11(b)).

In future work, we plan to investigate how the fuzzy set operations can be applied to represent the other shape relations, and also the relations among different modalities, e.g. images and 3D shapes.

Acknowledgements We thank the anonymous reviewers for their comments and suggestions. M. Sung acknowledges the support in part by the Korea Foundation for Advanced Studies. A. Dubrovina acknowledges the support in part by The Eric and Wendy Schmidt Postdoctoral Grant for Women in Mathematical and Computing Sciences. L. Guibas acknowledges the support of NSF grants IIS-1528025, DMS-1521608, and DMS-1546206, and gifts from the Adobe systems, Amazon AWS and Autodesk corporations.

References

[BWCB11] Bordes A., Weston J., Collobert R., Bengio Y.: Learning structured embeddings of knowledge bases. In *AAAI* (2011). 3

[CFG*15] Chang A. X., Funkhouser T. A., Guibas L. J., Hanrahan P., Huang Q., Li Z., Savarese S., Savva M., Song S., Su H., Xiao J., Yi L., Yu F.: Shapenet: An information-rich 3d model repository. *CoRR abs/1512.03012* (2015). 1, 5

[CKGF13] Chaudhuri S., Kalogerakis E., Giguere S., Funkhouser T.: AttribIt: Content creation with semantic attributes. In *UIST* (2013). 2

[CKGK11] Chaudhuri S., Kalogerakis E., Guibas L., Koltun V.: Probabilistic reasoning for assembly-based 3d modeling. In *SIGGRAPH* (2011). 2

[DQN17] Dai A., Qi C. R., Niessner M.: Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *CVPR* (2017). 9

[FKS*04] Funkhouser T., Kazhdan M., Shilane P., Min P., Kiefer W., Tal A., Rusinkiewicz S., Dobkin D.: Modeling by example. In *SIGGRAPH* (2004). 1, 2

[HYL17] Hamilton W., Ying Z., Leskovec J.: Inductive representation learning on large graphs. In *NIPS*. 2017. 3

[KCKK12] Kalogerakis E., Chaudhuri S., Koller D., Koltun V.: A probabilistic model for component-based shape synthesis. In *SIGGRAPH* (2012). 2

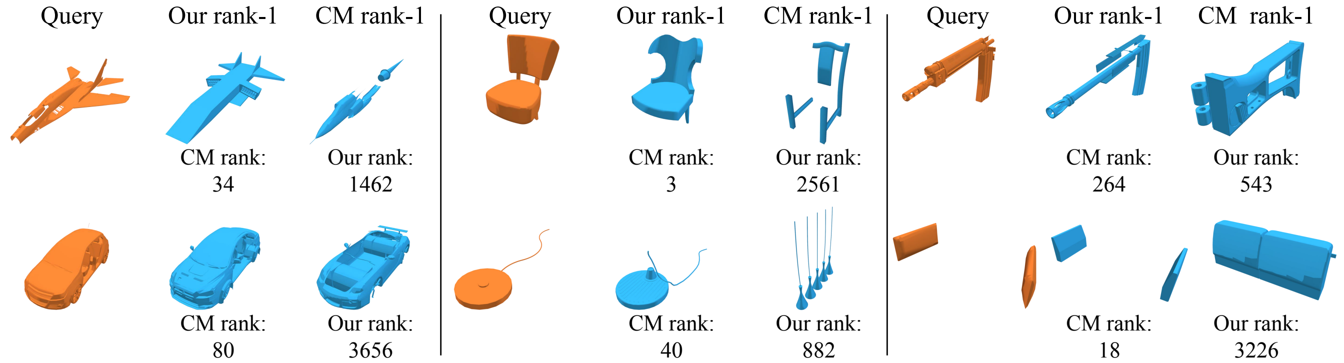


Figure 9: *Comparison of rank-1 nearest neighbors detected using our interchangeability measure, and distances in the embedding space of ComplementMe [SSK*17]. The numbers below show the ranks when using different methods. ComplementMe sometimes finds wrong interchangeable partials shapes while ours correctly discriminates correct and wrong interchangeable shapes.*

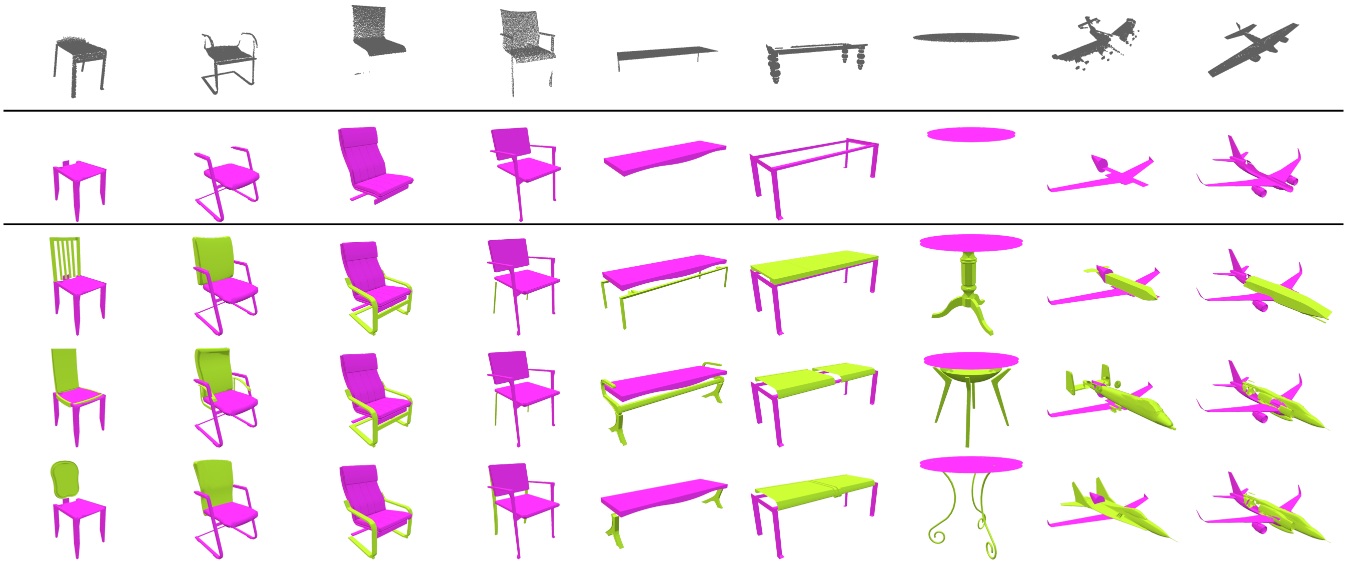


Figure 10: *Examples of partial scan completion. We used synthetic partial scan data from the benchmark of [SKAG15] (first row). We first use ICP to retrieve the best fitting partial shape from our database (second row, colored in pink), and then complete it to a full shape using various complements retrieved by the proposed method (colored in green). Three out of the best ten completions are shown in third, fourth, and fifth rows. The positions of the complements are automatically predicted using the placement network of [SSK*17].*

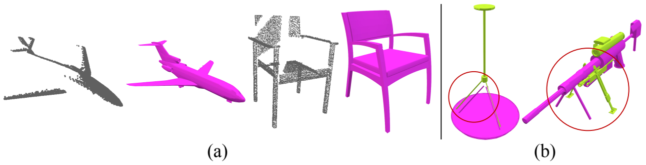


Figure 11: *Limitations of our method. (a) Our complement retrieval is not applicable to complete scans when they include all components but have small missing areas. Left is a scan, and right is a ICP retrieval result as done in Figure 10. (b) Small or thin parts may not be recognized well due to the limited resolution of point clouds, and thus the retrieved complements may include duplicate parts with the query (circled in red).*

[KFF15] Karpathy A., Fei-Fei L.: Deep visual-semantic alignments for generating image descriptions. In *CVPR* (2015). 3, 5

[KLM*13] Kim V. G., Li W., Mitra N. J., Chaudhuri S., DiVerdi S., Funkhouser T.: Learning part-based templates from large collections of 3d shapes. In *SIGGRAPH* (2013). 1

[KLSW15] Klein B., Lev G., Sadeh G., Wolf L.: Associating neural word embeddings with deep image representations using fisher vectors. In *CVPR* (2015). 6

[KW17] Kipf T. N., Welling M.: Semi-supervised classification with graph convolutional networks. In *ICLR* (2017). 3

[LSQ*15] Li Y., Su H., Qi C. R., Fish N., Cohen-Or D., Guibas L. J.: Joint embeddings of shapes and images via cnn image purification. In *SIGGRAPH* (2015). 3

[LXC*17] Li J., Xu K., Chaudhuri S., Yumer E., Zhang H., Guibas L.: Grass: Generative recursive autoencoders for shape structures. In *SIGGRAPH* (2017). 1

[MSC*13] Mikolov T., Sutskever I., Chen K., Corrado G., Dean J.: Distributed representations of words and phrases and their compositionality. In *NIPS* (2013). 3

[NK17] NICKEL M., KIELA D.: Poincaré embeddings for learning hierarchical representations. In *NIPS* (2017). 3

[PARS14] PEROZZI B., AL-RFOU R., SKIENA S.: Deepwalk: Online learning of social representations. In *KDD* (2014). 3

[QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR* (2017). 5, 6

[SCMN13] SOCHER R., CHEN D., MANNING C. D., NG A. Y.: Reasoning with neural tensor networks for knowledge base completion. In *NIPS* (2013). 3

[SFCH12] SHEN C.-H., FU H., CHEN K., HU S.-M.: Structure recovery by part assembly. In *SIGGRAPH Asia* (2012). 1, 2

[SKAG15] SUNG M., KIM V. G., ANGST R., GUIBAS L.: Data-driven structural priors for shape completion. In *SIGGRAPH Asia* (2015). 1, 2, 9, 10

[SKP15] SCHROFF F., KALENICHENKO D., PHILBIN J.: Facenet: A unified embedding for face recognition and clustering. In *CVPR* (2015). 3, 5

[SMKLM15] SU H., MAJI S., KALOGERAKIS E., LEARNED-MILLER E.: Multi-view convolutional neural networks for 3d shape recognition. In *ICCV* (2015). 8, 9

[SSK*17] SUNG M., SU H., KIM V. G., CHAUDHURI S., GUIBAS L.: Complementme: Weakly-supervised component suggestions for 3d modeling. In *SIGGRAPH Asia* (2017). 1, 2, 3, 5, 6, 8, 9, 10

[Tri17] TRIMBLE: 3d warehouse, 2017. URL: https://3dwarehouse.sketchup.com/. 1

[VKFU16] VENDROV I., KIROS R., FIDLER S., URTASUN R.: Order-embeddings of images and language. In *ICLR* (2016). 3, 4, 5, 6

[vKXZ*13] VAN KAICK O., XU K., ZHANG H., WANG Y., SUN S., SHAMIR A., COHEN-OR D.: Co-hierarchical analysis of shape structures. In *SIGGRAPH* (2013). 2

[WZX*16] WU J., ZHANG C., XUE T., FREEMAN W. T., TENENBAUM J. B.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NIPS* (2016). 3

[XZCOC12] XU K., ZHANG H., COHEN-OR D., CHEN B.: Fit and diverse: Set evolution for inspiring 3d shape galleries. In *SIGGRAPH* (2012). 2

[YKC*16] YI L., KIM V. G., CEYLAN D., SHEN I.-C., YAN M., SU H., LU C., HUANG Q., SHEFFER A., GUIBAS L.: A scalable active framework for region annotation in 3d shape collections. *SIGGRAPH Asia* (2016). 2, 8, 9

[Zad65] ZADEH L.: Fuzzy sets. *Information and Control* (1965). 2, 4

[Zim01] ZIMMERMANN H.-J.: *Fuzzy Set Theory - and Its Applications*. Springer, 2001. 4

Appendix

Proposition 1 $\max\left(E_c(x \to z), E_c(y \to z)\right) \leq E_c((x \vee y) \to z)$

Proof

$$\begin{aligned}
&\max\left(E_c(x \to z), E_c(y \to z)\right)\\
&= \max\left(\sum_i \max(0, f(x)_i - g(z)_i)^2, \sum_i \max(0, f(y)_i - g(z)_i)^2\right)\\
&\leq \sum_i \max\left(\max(0, f(x)_i - g(z)_i)^2, \max(0, f(y)_i - g(z)_i)^2\right)\\
&= \sum_i \max\left(0, \max(f(x)_i, f(y)_i) - g(z)_i\right)^2\\
&= E_c((x \vee y) \to z)
\end{aligned}$$

□

Proposition 2 $\max\left(E_c(z \to x), E_c(z \to y)\right) \leq E_c(z \to (x \wedge y))$

Proof

$$\begin{aligned}
&\max\left(E_c(z \to x), E_c(z \to y)\right)\\
&= \max\left(\sum_i \max(0, f(z)_i - g(x)_i)^2, \sum_i \max(0, f(z)_i - g(y)_i)^2\right)\\
&\leq \sum_i \max\left(\max(0, f(z)_i - g(x)_i)^2, \max(0, f(z)_i - g(y)_i)^2\right)\\
&= \sum_i \max\left(0, f(z) - \min(g(x)_i, g(y)_i)\right)^2\\
&= E_c(z \to (x \wedge y))
\end{aligned}$$

□