Computers & Graphics (2023)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Analyzing part functionality via multi-modal latent space embedding and interweaving

Jiahao Cui[a], Shuai Li[a,b], Fei Hou[c,d,*], Aimin Hao[a], Hong Qin[e,**]

[a] *State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, 100191, China*
[b] *Zhongguancun Laboratory, Beijing, 100191, China*
[c] *State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, 100190, China*
[d] *University of Chinese Academy of Sciences, Beijing, 100049, China*
[e] *Department of Computer Science, Stony Brook University (SUNY at Stony Brook), Stony Brook, New York 11794-2424, USA*

ARTICLE INFO

Article history:
Received June 17, 2023

Keywords: Part functionality analysis, latent space, cross-domain hybrid, functionality-aware shape modeling

ABSTRACT

In this paper, we advocate a novel method for analyzing the functionality of parts in 3D objects. In contrast to prior research, our method no longer characterizes the functionality of an object part using its single type of qualities (or attributes), e.g., geometry or interactions, nor by weighing the significance of various qualities. Instead, we consider the latent space of part functions as a semantic feature space comprehensively defined by part qualities. To learn such a space by parameterizing and encoding semantic features from multi-channel, we begin by learning multi-modal latent space using shapes, textures, and interaction scenes. Next, the latent space of part functions is generated by embedding and interweaving these multi-modal spaces into a space with a higher dimension. We devise loss functions to direct the embedding and interweaving of multi-modal spaces while preserving their manifolds. Consequently, the learned functionality latent space can capture the similarities between semantic features related to functionality and encode them into high-level functional representations. We assess this innovative approach on diverse categories of textured 3D shapes. Extensive experiments have exhibited our method's parametric and encoding capability towards functionality-centric shape analysis and synthesis, including shape functionality analysis, functionally-similar shape retrieval, and functionality-aware modeling, all of which are of the essence to new graphics techniques and applications.

© 2023 Elsevier B.V. All rights reserved.

1. Introduction

With the swift advancement in both acquisition hardware and 3D modeling software, the number of digitized 3D models increases at an exponential pace in the most recent decade. There has been a growing need for new algorithms and tools that can facilitate shape analysis and synthesis. Towards this goal, a variety of shape analysis approaches have been proposed, including methods for feature detection, shape segmentation, and shape completion [1, 2, 3, 4]. The majority of the previous methods concentrate on pure geometric analysis, mainly relying on techniques that compute global/local features. Despite their popularity in processing 3D shapes, current methods oftentimes can provide little information about objects' functionality.

Functionality typically refers to how an object could be utilized [5, 6] and is a type of semantic feature characterized by the functions of parts. The formal, mathematically rigorous definition of the functionality of parts in 3D objects would provide a potent, high-level functional descriptor for shape synthesis and simulation. 3D parts' functions are generally characterized by their appearances, materials, and interactions with surrounding

*Corresponding author
**Corresponding author: Tel.: +1-631-632-8450; fax: +1-631-632-8384;
e-mail: houfei@ios.ac.cn (Fei Hou), qin@cs.stonybrook.edu (Hong Qin)

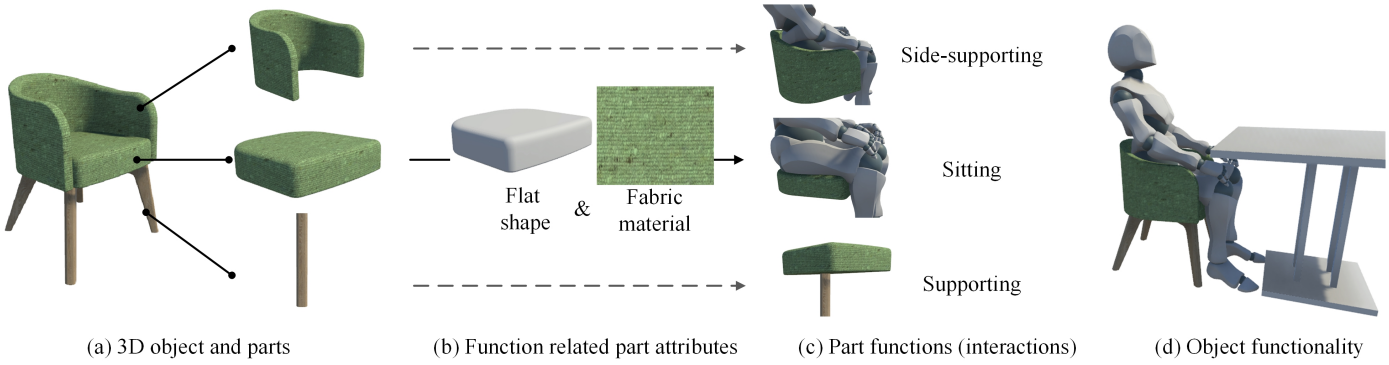

Fig. 1. The central methodology of our work is that part functionality is a high-level semantic feature visualized by interaction scenes, which can be deduced from the geometric and material properties of parts. (a): A real-world, man-made 3D object must serve its designed functional purpose and be composed of parts with specific functions. (b): Parts are highly connected in their various properties, and (c): the objective rule about the functions of parts lies behind the co-occurrence of these properties. (d): The results of part functionality analysis based on multi-modal data contribute to a thorough comprehension of the 3D object's functionality.

parts or objects. We call these qualities, which are important cues to infer the functionality of object parts, function cues. Ongoing pursuits in functional shape analysis have obtained functional representations from the aforementioned types of function cues [7]. Some methods study static and dynamic spatial relationships between object subcomponents [8, 9, 10, 11]. In a similar spirit, recent works utilize interaction contexts (ICON) to study functionality [12, 13, 14]. While the aforementioned methods have demonstrated that incorporating information at a more semantic level could greatly benefit functionality-centric shape analysis, there are still obstacles to fusing function cues into a comprehensive functional representation. The majority of previous research either implies that interactions define functionality or requires extensive semantic labeling to train the analysis model, which hinders the discovery of all possible functions. In addition, existing methodologies tend to combine function cues in a weighted summation fashion and do not provide information regarding semantic relations between different cues, rendering cross-domain part modeling impossible.

We introduce a novel method that aims to learn the functionality latent space of 3D object parts. Our method is highly motivated by the observation that humans predict functionality by analyzing semantic information across qualities from multi-channel sources, as shown in Figure 1. It is our belief that real-world 3D shapes, especially man-made 3D object components, are highly connected in their various characteristics (e.g., the correlation between shape and interaction). The objective rule about the usage of object parts lies behind the co-occurrence of these characteristics. Consequently, part functionality is a high-level semantic description that is extracted from and made palpable by the part qualities. We consider different part qualities as projections of the functionality latent space in lower-dimensional and multi-modal data spaces to formulate such a bidirectional mapping relation. Our goal is to infer the latent space from data spaces, explore cross-modal relationships of part qualities, and finally parameterize part functions.

However, learning such a space is not feasible due to the limited training data against an expected semantic feature space. This motivates us to implicitly model the manifolds of data spaces. Following the common manifold assumption, we assume that part quality data lie close to a low-dimensional and locally linear manifold in the data space. From the perspective of computer graphics, we use multi-modal data consisting of shapes, textures, and interaction scenes to analyze part qualities and encode multi-channel semantic features. Utilizing manifold and multi-modal data facilitates not only the parameterization of function cues using limited unlabeled data, but also the cross-domain measurement and synthesis of functions by utilizing meaningful Euclidean distances in manifolds.

Another challenge in learning the latent space is finding projection functions that allow for highly nonlinear representations between the latent variables of semantic features and the multi-modal data. The functions are supposed to fuse multi-modal data, as well as respect the underlying distribution implied by their own latent spaces. We address this challenge by devising a new learning method, latent space embedding and interweaving, which warps, stretches, and concatenates low-dimensional latent space in high-dimensional space under the guidance of explicit constraints. In particular, we first learn shape, texture, and interaction latent spaces. The learned spaces are then embedded into a joint latent space. During the embedding, we use shape-texture-interaction triplets of model parts as junctions to interweave latent spaces. The embedding and interweaving fuse data at the semantic level via metric learning.

To the best of our knowledge, our work develops the first method considering materials for the function analysis of 3D shapes. Compared with the previous works, our method goes beyond inferring functionality entirely from the interaction context. Using textures facilitates comprehensive analysis of part and object functionality. Moreover, our method brings the understanding of functionality near to the semantic level by identifying the inherent relations, i.e. semantic associations, between shape, material, and interaction. In particular, these multi-modal data are clustered by functional similarities in functionality space. This enables the discovery of multiple functions of an object part without using semantic identifiers as input, a problem that previous methods were unable to solve.

The latent space can provide high-level and powerful functionality representations, which benefit from manifold learning and fusing multi-modal data and can serve different applications. We perform a number of experiments to verify our method's parametric and encoding capability toward-

s functionality-centric shape analysis and synthesis. We also demonstrate in our evaluation that our method provides a more accurate and comprehensive description of part functionality when compared to the state-of-the-art functionality analysis approaches. The salient contributions of this paper include:

- We articulate a novel method for functionality analysis using multi-modal latent space embedding and interweaving.
- Through extensive experiments, we showcase that the learned latent space can encode different function cues into high-level functionality representations.
- We evaluate our method and draw a contrast between our method and the state-of-the-art approaches such as the ICON and the fSIM-NET and validate our method's advantages in functionality-centric shape analysis and synthesis.

2. Related work

2.1. Multi-modal learning

Multiple domains are the source of high-level, semantic information. There have been many applications of multi-modal learning, including text-to-image synthesis, medical applications with spatial-temporal data [15], and robotics applications with visual and depth data [16]. These data from various channels are semantically correlated and offer complementary information. To capture the 'concept' in the real world with such data, previous works have focused on learning joint representations by fusing multi-modal data. Huiskes et al. [17] enhanced the accuracy of image classification by incorporating captions and conventional low-level image features. A similar approach by Guillaumin et al. [18], which is based on multiple kernel learning, demonstrates that an additional text modality can improve the accuracy of various object recognition tasks. A second class of methods learns joint representations by extending Canonical Correlation Analysis [19]. However, these methods do not scale well to large amounts of unlabeled data.

On the deep learning side, deep autoencoders or deep Boltzmann machines have been used to learn joint representations. Ngiam et al. [20] introduced a stacked multimodal autoencoder to learn joint audio and video representations. Srivastava and Salakhutdinov [21] trained a deep Boltzmann machine to model images and text tags. Sohn et al.[22] presented a generative model of multi-modal data by hallucinating the missing data modality and minimizing the variation of information. More recent work in text-image joint embedding [23] focuses on decoding distributed representations and using unsupervised learning. Kim et al. [24] proposed to discover cross-domain relations with generative adversarial networks. These approaches are similar in a common strategy to learn representations shared across multiple modalities at the higher layer of the deep network after learning layers of modality-specific networks. As a result, the learning algorithms are easily tempted to learn dominant patterns in each data modality separately while giving up learning patterns that coincide in multiple modalities.

Recent research on manifold learning enables the discovery of datasets' intrinsic features and the visualisation of their underlying manifolds. Moreover, manifold alignment is frequently used to establish correspondences between datasets [25]. Our method constructs joint descriptors of function cues by interweaving multiple latent spaces. We develop a latent space for multi-modal data fusion that goes beyond simple domain alignment. Due to the manifold's characteristics, such as its local resemblance to Euclidean space, the joint descriptors are discriminative and generative.

2.2. Functionality analysis of 3D shapes

The availability of 3D shape repositories and the rising number of applications that leverage geometric data necessitate the development of techniques for the functional analysis and functionality-oriented manipulation of 3D shapes. In retrospect, numerous past works on structure-aware analysis[26, 27] or data-driven shape analysis[28, 29] are related to functional analysis. Despite the fact that these connections are insufficient for comprehending the functionality of shapes, a number of these approaches serve as the foundation for later methods that perform explicit functionality analysis[7]. In these methods, the structure of a shape is typically represented by a multi-branch tree or a graph in order to identify symmetric[30] or recurring part arrangements[31]. Symmetry and recurrence are relevant to functionality because symmetric or recurrent parts have a tendency to perform identical functions. Nevertheless, merely detecting part arrangements does not reveal part functions. Fish et al.[10] analyzed the configurations of parts, encoded by multiple relations between parts such as angles and relative proportions. Yumer and Kara[32] abstracted shapes as simple primitives, and then characterized shapes by capturing the common configuration of these primitives. Hu et al.[11] developed a part mobility model that can predict spatial configuration and motion of object parts. Similarly, at the level of parts, Xu et al[33]. proposed PartAfford, a supervised learning framework for decomposing 3D shapes into parts and classifying those parts according to their affordance category. The primary limitation of these methods is that not all qualities of shape parts are related to functionality; these qualities would need to be filtered out in order to create a pure model of functionality[7].

Recent efforts have shifted towards encoding the high-level semantic information of functionality through the acquisition of additional knowledge. Zhao et al.[9] took into account the context of static 3D scenes and used Interaction Bisector Surface (IBS) to describe spatial relationships between shapes. IBS is typically insensitive to the fine-grained geometry of shapes. In another line of work, functional representations are derived from interaction properties captured using human agents or physical simulation particles. Given a shape, Kim et al.[34] predicted human pose and contact points. On the basis of predicted pose, a particular function can be assigned to the shape. Hinkle and Olson[35] introduced physical simulation-obtained features for predicting functionality. A group of spheres is dropped from above onto a shape, and the final distribution of spheres resulting from the simulation is used for functional classification. Using character animations, Pirk et al.[36] expanded the variety of particles. However, predefined human poses and physical simulations lack the complexity of real-world interactions, limiting the applicability of these approaches in a broader sense.

Research employing interaction contexts[12, 13, 37, 14] assumes that interactions define the functionality, which hinders

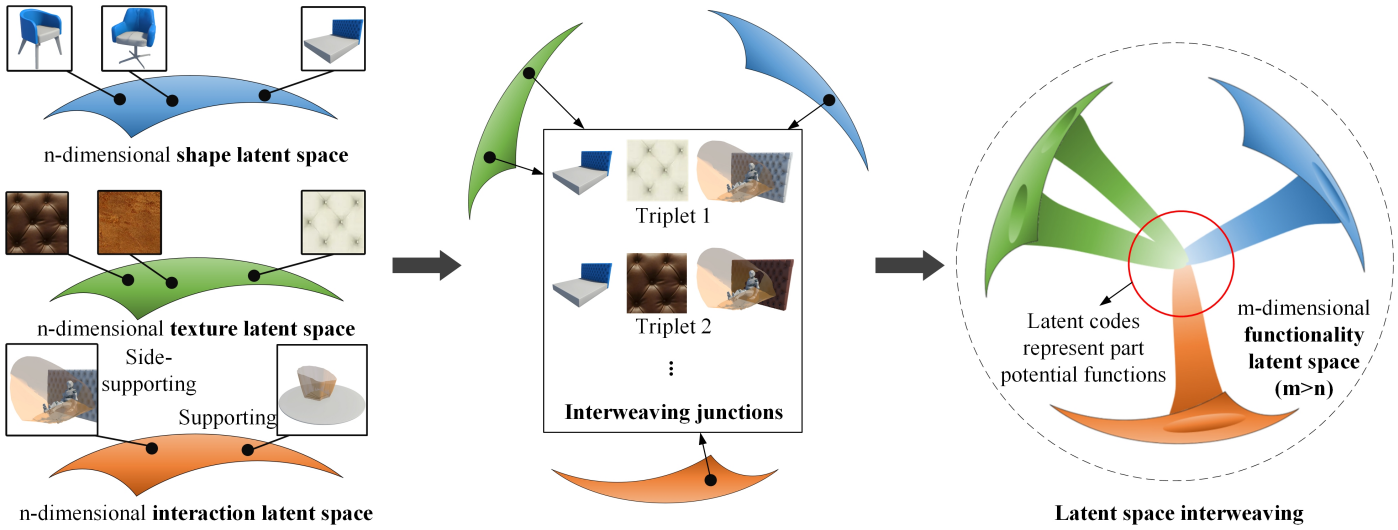

Fig. 2. Illustration of multi-modal latent space embedding and interweaving. Individual autoencoder-decoders are used to learn multi-modal latent spaces. Each latent space is a low-dimensional data space, and each function cue has a corresponding coordinate in its latent space. The combination of a textured shape and its interaction scene is regarded as a triplet of the form (Shape, Texture, Interaction) and is used as an interweaving junction. In order to obtain functionality latent space, multi-modal spaces are embedded within a higher-dimensional space and local regions near junctions are stretched together.

the encoding of all pertinent semantic information in order to discover all possible functions. In contrast to previous work, our learning method brings the semantic properties of shapes closer to the comprehension of functionality. This distinction is necessary for building the semantic similarity between function cues from different modal data. We demonstrate that our method is capable of measuring functional similarities between part qualities and predicting functionality. In addition, none of the preceding works addressed the issue of modifying or synthesising the function cues of model parts.

3. Overview of novel method

Definition of part functionality. We start from the mathematical definition of part functionality. In this work, we focus on static objects with fixed and non deformable parts. Two widely accepted and interrelated psychological theories [38] have been proposed regarding how humans perceive and comprehend 3D objects. According to one theory, humans perceive 3D objects based on their properties. According to the alternative theory, humans perceive 3D objects based on their part functions. Combining these two theories, we can define the function of an object part as follows:

$$\mathbf{z} = f(\mathbf{s}, \mathbf{x}, \mathbf{y}), \tag{1}$$

where $\mathbf{z}$ is the latent variable of part function, and $\mathbf{s}$, $\mathbf{x}$, $\mathbf{y}$ are the parameterized representations of shape, material and interaction features of an object part, respectively. f represents the process of analyzing and fusing function cues in order to reason about the function of parts.

The idea of part function definition underlying Equation 1 is that, on the one hand, although functionality as a semantic feature is difficult to parameterize, it can be achieved with the aid of parameterizable part attributes if a mapping can be established for the reasoning process, as shown in Figure 1. On the other hand, the attributes of parts, such as shape and material, are the basis of part functions, whereas interaction instances are explicit functional representations and the result of function reasoning based on shape and material. However, an interaction instance reflects a function but does not equal functionality because a part may have multiple interactions and functions, so it is necessary to construct a co-embedding semantic feature space on top of different function cues, which can represent the complementary functionality-related semantic information.

Function cue representations. From the perspective of computer graphics, we represent part shape with point clouds and part material using textures. Texture images frequently depict materials (such as wood, fabric, etc.) that are highly correlated with functions, whereas the material of a part cannot be deduced from its shape alone. Texture images can provide semantic descriptions for complex shape analysis and human-object interaction tasks. For the representation of interaction, we employ IBS[9], which has been extensively demonstrated to be capable of describing interactions between objects.

System components and their organization. Our primary objective is to leverage the latent space of concrete multi-modal data during the establishment of semantic-level space. Specifically, as shown in Figure2, we assume that each part function defined by Equation1 possesses its own coordinate in the functionality latent space, which is determined by shapes, textures, and interactions. In data spaces, the coordinates of shapes, textures, and interactions exist in their respective spaces. To fit this bidirectional mapping and explicitly construct functionality latent space from function cues, we propose the multi-modal latent space embedding and interweaving technique. Using individual autoencoder-decoders(Section 4), we first learn shape latent space, texture latent space, and interaction latent space. Then, low-dimensional latent spaces are embedded into a high-dimensional joint space(Section 5). The extra dimensions permit the deformation and concatenation of spaces. During the

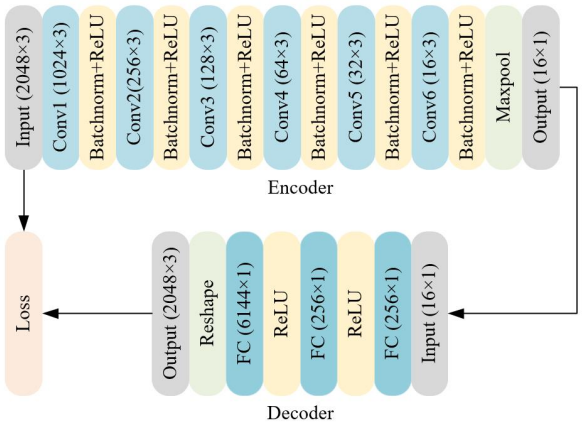

Fig. 3. The autoencoder-decoder networks architecture we used to learn shape latent space and interaction latent space.

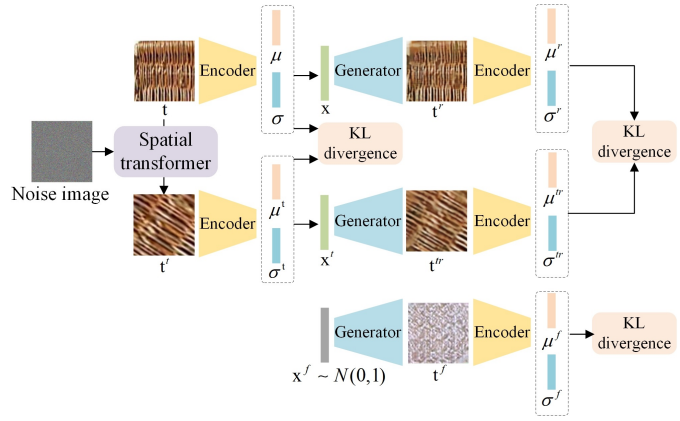

Fig. 4. The networks we used to learn texture latent space. We use a spatial transformer module to provide random affine transformation for inputs.

final phase of interweaving, local regions near junctions are stretched and converge to form a new subspace. The subspace combines multimodal data and specifies part functionality.

4. Function cue latent space

Since we use geometry data and texture images as function cue representations and establish a high-dimensional parameterization space with these multi-modality data, we will first describe how we use autoencoder-decoder networks to discover their distributions and latent spaces.

4.1. Learning shape latent space

We use a point cloud with 2048 points to represent a part shape. The encoder architecture for point cloud follows the design principle of PointNet [39] to encode every point independently: 1-D convolutional layers and an increasing number of features. As shown in Figure 3, we use six convolutional layers, each followed by a ReLU and a batch-normalization layer. The output vectors implicitly define a parameter space. The decoder consists of three fully connected layers and two ReLU layers. We use the Chamfer Distance [40] as the loss to train the autoencoder-decoder network. We experimented with different dimensions for the output vectors (8, 16, 32), and we found that 16 dimensions are enough for reconstructing point clouds and representing part shapes. Instead, lower-dimensional vectors tend to lead to insufficient geometry details, and higher-dimensional vectors will make it challenging to find an optimal solution for multiple spaces interweaving.

4.2. Learning texture latent space

Previous image works have focused on learning the generative model or manifold of indoor pictures[41] and natural texture images[42]. In addition to the typical requirements for learning a latent space, the spatial transform must also be considered in our research. Texture images from various parts may have significantly distinct structures and particulars. A texture image of fabric, for instance, typically consists of repeating high-frequency details. In contrast, natural textures, such as wood, must have a wide range of scale variations. These spatial transforms are strongly related to the local data distribution in the learned latent space and influence parameterization. Nevertheless, it is challenging to encode transform information or learn their data distribution using conventional autoencoders, particularly when training data consists of sparse samples and transform information is not explicitly presented.

We address this problem by combining IntroVAE [41] with Spatial Transformer Networks [43]. As shown in Figure 4, we use a spatial transformer module to provide random affine transformation for training data. The motivation behind this network design is to generate infinite variations from sample images. The outputs of the spatial transformer module fill the data domain with infinite pixel transform and facilitate data distribution learning. Given a sample t as input, the encoder is designed to output two individual variables, μ and σ. The input x of the generator is sampled from $N(x;\mu,\sigma^2)$ using reparameterization: $x = \mu + \sigma \cdot \varepsilon$ where $\varepsilon \sim N(0,1)$. Given N origin images and their transformed images, we use KL-divergence as a loss term to encourage the encoder to match data distributions:

$$\begin{aligned}\mathcal{L}_{KL-r}(t) = &-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{d_x}({\mu_{ij}}^2 + {\sigma_{ij}}^2 + {\mu^t}_{ij}^2 + {\sigma^t}_{ij}^2) \\ &+\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{d_x}(2 + \log((\sigma_{ij}{\sigma^t}_{ij})^2)),\end{aligned} \tag{2}$$

where d_x is the dimension of the latent codeword x, μ^t and σ^t are the latent variables of the transformed image t^t. For samples generated from noise code x^f and their latent variables μ^f and σ^f, the KL-divergence can be computed as:

$$\mathcal{L}_{KL-f}(x^f) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{d_x}(1 + \log({\sigma^f}_{ij}^2) - {\mu^f}_{ij}^2 - {\sigma^f}_{ij}^2). \tag{3}$$

Following the original IntroVAE [41] network structure which is in a similar manner to GANs, the total loss functions for encoder and generator are defined as:

$$\begin{aligned}\mathcal{L}_E = &\max(0, m - \mathcal{L}_{KL-r}(t^r)) + \max(0, m - \mathcal{L}_{KL-f}(x^f)) + \\ &\alpha\mathcal{L}_{KL-r}(t) + \beta\sum_{i=1}^{N}\|t_i - {t^r}_i\|^2,\end{aligned} \tag{4}$$

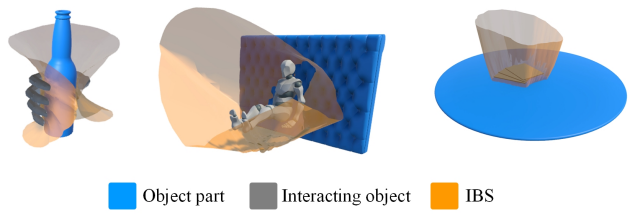

Fig. 5. Different types of interactions and IBS. From left to right: holding, side-supporting, and supporting. Interaction scenes in our dataset are constructed considering three part-centric aspects: interaction with a human hand, a human body, and interaction with another 3D part or object.

$$\mathcal{L}_G = \mathcal{L}_{KL-r}(t^r)) + \mathcal{L}_{KL-f}(x^f)) + \beta \sum_{i=1}^{N} \|t_i - t^r{}_i\|^2, \tag{5}$$

where t^r is the image reconstructed from the latent code of sample t, m is a positive margin, and α and β are weighting parameters used to balance the importance of each item. The network architectures of the convolution encoder and generator are the same as the original IntroVAE.

4.3. Learning interaction latent space

IBS represents a spatial region between two shapes and captures part interaction features (Figure 5), such as enclose/holding, sitting, supporting, etc. Given two sets of points P_1 and P_2, sampled from two shapes S_1 and S_2, respectively, the IBS between S_1 and S_2 is defined as the set of points equidistant to both P_1 and P_2. The IBS can be obtained by computing the 3D Voronoi diagram of all the sample points and selecting only the surface of the diagram that bisect points from two shapes. Since the IBS is infinite, we truncate it using a bounding sphere of S_1 and S_2.

As shown in Figure 5, the topological and geometry features of IBS reflect interaction types and details. Previous works[9, 12, 13] use Betti number [44] or point feature histogram to describe the geometry and topological features of IBS. These statistical descriptors with cluster algorithms are then used to differentiate between interactions. However, it is challenging to reconstruct IBS from statistical descriptors considering a series of summations in histogram computation.

Motivated by the prevalence of point cloud encoders and surface reconstruction algorithms, we encode IBS with latent codes after parameterizing them with their point clouds. The topological characteristics of IBS can be extracted and represented using the global characteristics of point clouds. Similarly, the local characteristics of point clouds, which are computed from the points and their neighbours, can precisely describe the geometry characteristics of IBS. Besides, latent codes facilitate the reconstruction of IBS through decoding, which is crucial for interaction visualisation. In our implementation, we triangulate the IBS surface faces before sampling a point cloud for each IBS. The point cloud is then used to train an autoencoder-decoder using the same architecture as shown in Figure 3.

5. Multiple latent space embedding and interweaving

With encoders, which encode shape, texture, and interaction, we learn the latent space of different modality function cues. We represent the learned spaces with the output latent codes of encoders, as $\mathbf{S} = \{s_1, s_2, ..., s_{N_s}\} \in \mathbf{R}^{d_s \times N_s}$, $\mathbf{X} = \{x_1, x_2, ..., x_{N_x}\} \in \mathbf{R}^{d_x \times N_x}$, and $\mathbf{Y} = \{y_1, y_2, ..., y_{N_y}\} \in \mathbf{R}^{d_y \times N_y}$, where the columns s_i are shape latent codes, the columns x_i are texture latent codes, and y_i are interaction latent codes. The dimensions of the latent codes are d_s, d_x, and d_y. N_s, N_x, and N_y are the numbers of sample data. The goal of the embedding and interweaving is to obtain a latent space $\mathbf{Z} = \{z_1^s, z_2^s, ..., z_{N_s}^s, z_1^x, z_2^x, ..., z_{N_x}^x, z_1^y, z_2^y, ..., z_{N_y}^y\} \in \mathbf{R}^{d_z \times (N_s+N_x+N_y)}$ for part functions, where z_i^s, z_i^x and z_i^y are d_z dimensional latent codes which represent the coordinates of function cues in the functionality latent space, and d_z is the dimension of the functionality latent space accordingly.

5.1. Algorithm

We use Multilayer Perceptrons (MLPs) to perform multi-modal latent space embedding and interweaving in an effort to identify mapping functions. Not only need to interweave junctions from multi-modal latent spaces, but the high-dimensional functionality latent space is also supposed to maintain local data distribution of the lower-dimensional latent space to benefit from manifold learning. Therefore, to achieve the above goals, we formulate the loss function for the embedding and interweaving as follow:

$$\mathcal{L}_{Total} = \mu_1 \mathcal{L}_C + \mu_2 \mathcal{L}_S + \mu_3 \mathcal{L}_R. \tag{6}$$

The first two terms represent the latent space interweaving error (Section 5.2) measured by distances between function cue latent codes after interweaving. The last term indicates the reconstruction error (Section 5.3) of the latent space local data distribution. μ_1, μ_2, and μ_3 control the relative importance of each term. We experimented with two strategies for interweaving three latent spaces. One is training three MLPs and embedding three latent spaces to target functionality latent space simultaneously. However, this straightforward solution easily leads to nonconvergence. Another way is to interweave two latent spaces and update the parameters of MLPs sequentially, as listed in Algorithm 1. By trial and error, we found the second strategy is better in training performance, and the loss functions are more efficient to compute.

5.2. Latent space interweaving error

We first introduce loss functions for embedding and interweaving two latent spaces $\mathbf{X}$ and $\mathbf{Y}$. We assume that the shape, the texture image, and the interaction scene of a part constitute a triplet, that their coordinates in original latent space can be embedded to the same point or neighbouring regions in functionality latent space, and that they can be used as interweaving junctions because they contribute to the same part function. As depicted in Figure 6 (a), to explicitly construct functionality latent space, local regions near junctions should congregate along directions of interweaving. Consequently, we penalise large separations between junctions after embedding and interweaving and formulate the interweaving error loss as follows:

$$\mathcal{L}_C(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{b_c} \|z_i^x - z_{ci}^y\|^2, \tag{7}$$

Algorithm 1 Main steps of learning functionality latent space.

Input: A collection of 3D part shapes with textures and interaction scenes, hyperparameters $\mu_1, \mu_2, \mu_3, k, \theta$, and iteration number t

Output: A functionality latent space $\mathbf{Z} = \{z_1, ..., z_{N_z}\} \in \mathbf{R}^{d_z \times N_z}$

1: Sample point clouds from 3D shapes and train encoder E_1 to learn shape latent space $\mathbf{S} = \{s_1, ..., s_{N_s}\} \in \mathbf{R}^{d_s \times N_s}$ (Section 4.1)
2: Extract texture images and train encoder E_2 to obtain the latent space of textures: $\mathbf{X} = \{x_1, ..., x_{N_x}\} \in \mathbf{R}^{d_x \times N_x}$ (Section 4.2)
3: Compute IBS with interaction scenes and sample point clouds from IBS
4: Train encoder E_3 and obtain the latent space of interaction: $\mathbf{Y} = \{y_1, ..., y_{N_y}\} \in \mathbf{R}^{d_y \times N_y}$ (Section 4.3)
5: Select k neighbors for each function cue in their latent spaces and compute linear coefficients
6: Initialize MLPs
7: **for** iterator= $1, 2, 3, ..., t$ **do**
8: Embed $\mathbf{S}$ and $\mathbf{X}$ into high-dimensional latent space
9: Compute interweaving error $\mathcal{L}_C(\mathbf{S}, \mathbf{X})$ and $\mathcal{L}_S(\mathbf{S}, \mathbf{X})$ (Section 5.2)
10: Compute reconstruction error $\mathcal{L}_R(\mathbf{S}, \mathbf{X})$ (Section 5.3)
11: Optimize $\mathcal{L}_{Total}(\mathbf{S}, \mathbf{X})$ (Equation 6) and update MLPs
12: Repeat $Step8 - 11$ with input: $\mathbf{X}$ and $\mathbf{Y}$
13: Repeat $Step8 - 11$ with input: $\mathbf{Y}$ and $\mathbf{S}$
14: **end for**

where z_i^x and z_{ci}^y are the latent codes of texture x_i and interaction y_{ci}, respectively, after the embedding and interweaving. Texture x_i and the interaction y_{ci} are extracted from the same part. Therefore they make up a pair of corresponding junctions. b_c is the batch size of each training epoch. This loss function can be viewed as a hard constraint since it is enforced with very high weight in Equation 6.

Note that a texture image may appear in several parts that have the same material. This condition also applies to interactions. Consequently, a junction latent code we used may have more than one corresponding junction (Figure 6 (a)). These one-to-many situations distinguish our latent space interweaving operation from manifold alignment techniques. Latent codes will converge if they share a corresponding junction from the other latent space. As a result, data points near the junctions are involved in the convergence and give rise to dense data distribution in functionality latent space, as shown in Figure 6 (b). In accordance with the explicit constraint provided by Equation 7, interweaving occurs in both inner-space and cross-space. This could result in manifold collapse and render the learned functionality representation indistinguishable.

To avoid manifold collapse, we enlarge separations between data points in the learned functionality latent space, which can be seen as a scaling operation:

$$\mathcal{L}_S(\mathbf{X}, \mathbf{Y}) = -\frac{1}{{b_x}^2} \sum_{i=1}^{b_x} \sum_{j=1}^{b_x} \|z_i^x - z_j^x\|^2 - \frac{\theta}{{b_y}^2} \sum_{p=1}^{b_y} \sum_{q=1}^{b_y} \|z_p^y - z_q^y\|^2, \quad (8)$$

where θ is a hyperparameter used to balance and unify two latent spaces, b_x and b_y are batch sizes of each training epoch and are used to compute mean Euclidean distance between points in each latent space. We assign a lower weight to this constraint to eliminate its conflict with the interweaving procedure.

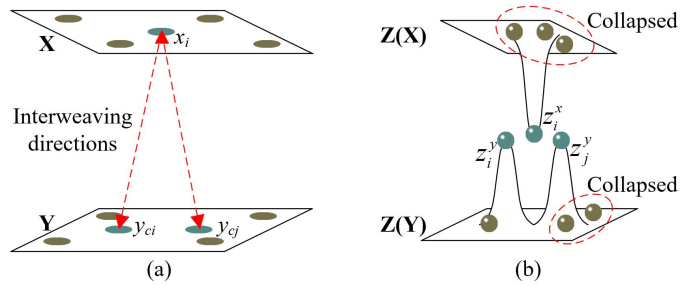

Fig. 6. Illustration of interweaving two latent spaces. (a): Given junctions x_i, y_i, and y_j, X and Y are supposed to embed into a higher dimensional latent space and local regions near the junctions should gather along with interweaving directions. (b): Interweaving result. Data points near the junctions will converge and cause manifold collapse.

5.3. Latent space reconstruction error

To ensure that the local data distribution of both latent spaces are preserved following embedding and interweaving, we use a reconstruction error, inspired by Locally Linear Embedding[45], to constrain neighbor relationships between the data points of individual local manifolds. As shown in Figure7, we consider a local region in a learned latent space that approximates a manifold. This region's data distribution is described by linear coefficients that reconstruct each data point from its neighbors. Given a data point x_i in the learned texture latent space, we first assign neighbors to it using the K-Nearest Neighbors. The linear coefficients can be computed by solving the following optimization problem:

$$\begin{aligned} &\arg\min_{\omega_{ij}} \|x_i - \sum_{j=1}^{k} \omega_{ij} x_j\|^2 \\ &s.t. \qquad \sum_{j=1}^{k} \omega_{ij} = 1, \end{aligned} \quad (9)$$

where x_j is the jth neighbor of x_i. The weight ω_{ij} summarizes the contribution of x_j to the reconstruction and reflects manifold properties of the local region close to x_i. Similarly, weights λ_{pq} which describe manifold properties in interaction latent space, can be obtained by solving the following problem:

$$\begin{aligned} &\arg\min_{\lambda_{pq}} \|y_p - \sum_{q=1}^{k} \lambda_{pq} y_q\|^2 \\ &s.t. \qquad \sum_{q=1}^{k} \lambda_{pq} = 1. \end{aligned} \quad (10)$$

We expect each data point and its neighbors to maintain their relationships in a locally linear region of the high-dimensional latent space. In particular, the same weights ω_{ij} that reconstruct x_i in low-dimensional manifold should also reconstruct its coordinates z_i^x in high-dimensional latent space, excluding new neighbors from another latent spaces. Therefore, reconstruction errors are measured by the cost function:

$$\mathcal{L}_R(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{b_x} \|z_i^x - \sum\nolimits_{ij} \omega_{ij} z_j^x\|^2 + \theta \sum_{p=1}^{b_y} \|z_p^y - \sum\nolimits_{pq} \lambda_{pq} z_q^y\|^2, \quad (11)$$

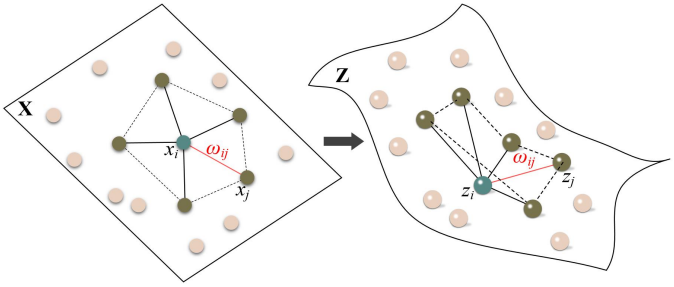

Fig. 7. Example of data distribution in a local region before (left) and after (right) embedding and interweaving . The data distribution of the local region can be characterized by linear coefficients that reconstruct each data point from its neighbors.

which adds up the squared distances between all the data points and their reconstructions. We assign lower weight to this constraint to eliminate its conflict with the interweaving.

6. Results and evaluations

In this section, we present experiment results, provide an evaluation of our method and discuss its performance against previous functionality analysis works.

6.1. Dataset and preparation

We collect a dataset of 1200 textured, man-made 3D shapes with synthetic interaction scenes. 3D shapes are collected from 3D Warehouse's high-quality database and produced by professional designers and studios. Designers utilized actual photographs as the basis for adding textures during the modeling. Textures are added to the parts shape through texture mapping. The association between textures and shapes conform to real-world patterns when real images are used as references. The 3D shapes we use cover 25 categories in the fSIM-NET dataset[14], which is an expanded version of the ICON dataset[12] and serves as a baseline in the state-of-the-art functionality analysis work. For the interaction scenes, we first follow the data preparation pipeline of the fSIM-NET dataset and construct scenes for each 3D shape. While the majority of 3D objects in their application scenarios will involve multiple interacting objects, which is a common scenario that has been studied in previous works, an object part typically has only one interaction type when performing its function. Therefore, we extract parts with their close-by interacting parts or external objects and split interactions into different types. In this manner, we compose interaction scenes that conform to our definition of part functionality and represent the functionality of objects involving multiple interactions. We perform a cross-validation evaluation where we divide shapes into a 9:1 training to test ratio. We use the shape-texture-interaction triplets of training set model parts as junctions to construct the latent space and use the test data for analysis, retrieval, and synthesis tasks. Dataset examples, comparison, and implementation details are provided in our supplementary material.

6.2. Functionality analysis results and comparison

We first show the capability of the learned latent space in functionality analysis. Part functionality analysis begins by encoding and mapping the shape and texture of the test object

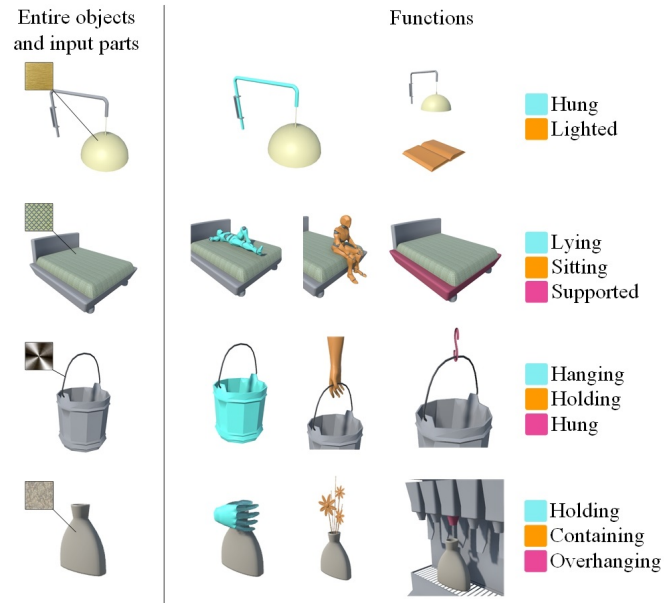

Fig. 8. Selected results of part functionality analysis using the functionality latent space. Left: Input parts (labelled with texture images) and the entire objects to which the input parts pertain. Right: Output functions represented by different interaction scenes.

part into the functionality latent space. In the local region of the latent space, the IBS samples adjacent to the mid-point of shape and texture are retrieved using Euclidean distance to represent the part's particular function. The indexed IBS and its corresponding interacting object are combined with the part to create a 3D scene, enabling visualization of the analysis results. Figure 8 depicts the functionality analysis results of object parts from the test set. The left column depicts the input part and the object to which it belongs. The middle columns depict the function analysis results (ranking the retrieved IBS according to Euclidean distance). It is evident from the results that all of the top-ranked interaction retrieval results are meaningful for the target 3D object part, e.g., a lampshade to illuminate a book or a mattress to sleep on. In addition to these interactions that represent conventional functions, our method can also investigate the potential functions of parts by analyzing the semantic information of shapes and materials, such as a mattress that can also be used for sitting due to its flat shape and soft fabric material and a vase can also be used to hold liquids.

On the basis of the function analysis of parts, the functionality analysis of 3D objects involving multiple interactions can be achieved by combining various types of interactions and the functions of different parts. As depicted in Figure 9(a), for a 3D object with multiple parts, we partition the parts and then retrieve IBS and interacting objects based on shape and material. Figure 9(b) depicts the 3D object's overall functionality as a result of its combination with various types of interactions.

The analysis of the 3D object's overall functionality permits comparison of our method with existing functionality analysis methods. Figure 10 displays the comparison results between our method and methods (ICON+LFD and fSIM-NET) proposed by Hu et al[12, 14]. From a large number of experimental results, four representative examples are selected. The two methods proposed by Hu et al. predict 3D object functions

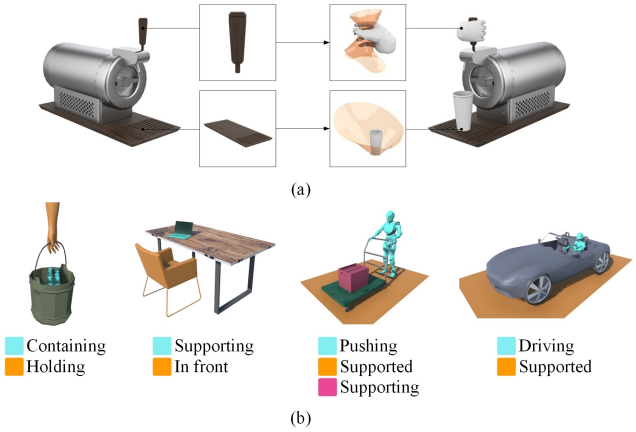

Fig. 9. 3D object functionality involving multiple interactions can be analyzed by combining various types of interactions and functions of parts.

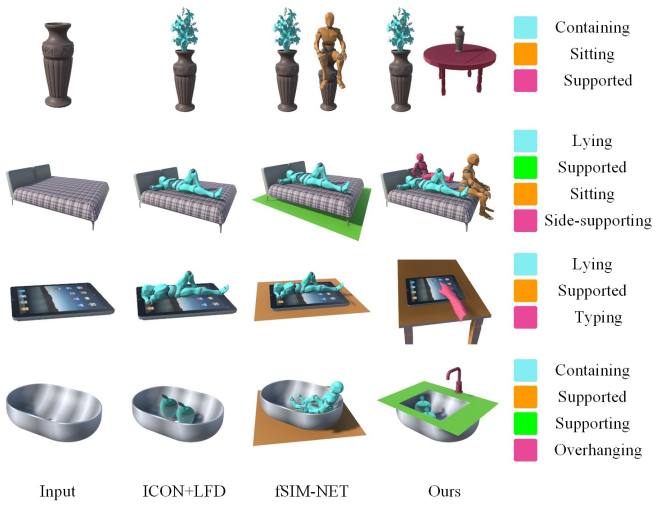

Fig. 10. Comparison of object functionality estimated by our method to those estimated by other methods.

via interaction relationship retrieval and interaction scene synthesis, and can therefore serve as a benchmark for our method. It is evident in Figure 10 that the first benchmark method can retrieve the regular functions of the 3D object, but is unable to identify all of the object's potential functions. The second benchmark method can generate novel scenes, such as a person sitting on a vase. This method uses only the geometric information of the 3D object, which is insufficient for functionality and can lead to the generation of illogical scenes. The fSIM-NET method migrates the interaction that a 3D object with similar geometry has to the target without comparing the material differences between them, and the interaction may not be adapted to the target object (lying on a pad and sitting in a sink, as shown in the 3rd and 4th row in Figure 10). In the real world, the material of an object influences how humans evaluate its function[13]. Even under certain conditions, the correct differentiation of functions can only be achieved by taking the material into account. Because material information is incorporated into the functionality analysis in the case of retrieving multiple interactions, our method guarantees that the predicted functions are reasonably reliable for both geometry and the material of the part. It should be noted that PartAfford[33] is the most recent work on part functionality. However, we did not conduct a comparison with it because PartAfford focuses on the task of functional part segmentation within the same shape category, whereas our analysis model allows for cross-category part functionality analysis. In addition, PartAfford is currently validated on only three interaction types and its application domains differ significantly from ours.

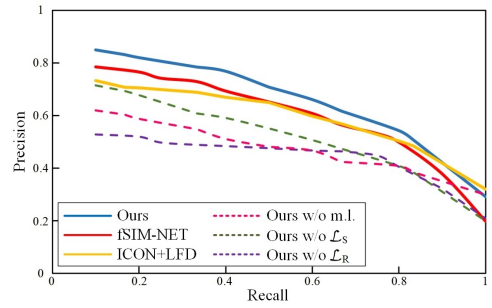

Fig. 11. Precision and recall diagram showing the performance of our method in interaction retrieval task, compared to the works of Hu al. [12, 14] and alternative learning strategies. "Ours w/o m.l." in the legend denotes learn the functionality latent space directly in one step with encoders and all loss functions, but without multi-modal latent space learning. Please refer to the text for details.

Figure11 displays the quantitative evaluation results of our method in the part function prediction task and the comparison results with other methods and alternative learning strategies. Experiments utilizing triplets in the test set retrieve 10 interactions using shape and texture for each triplet, verify that the retrieved interaction have the same type labels as those in the triplet, and then calculate precision and recall. Our method yields comparable results to the method proposed by Hu et al.[12, 14]. It demonstrates that despite the fact that our method explores more interaction types and potential functions using texture as a function cue, it still has high accuracy in retrieving the interaction and the explored potential functions are plausible. Existing methods do not utilize texture or material information for two primary reasons. One is that existing work uses shape category labels as function category labels and ignores texture materials. As a strong constraint, the label frequently makes it challenging for the algorithm to discover semantic information or functional similarities between categories, which can be captured by extracting material information. Existing methods on the other hand measure functions in different channel dimensions without establishing intrinsic correlations between channels, making it challenging to exploit semantic information. The qualitative and quantitative experiments demonstrate that our method can produce more precise and exhaustive functionality analysis results than the existing methods.

As shown in Figure 11, both removing loss function term $\mathcal{L}_S$ and $\mathcal{L}_R$ result in performance decrease. $\mathcal{L}_C$ directs local regions to converge while $\mathcal{L}_S$ directs the entire latent space to scale to be more discriminative, they operate at various levels of the latent space. By adjusting the weight parameters, all of these loss function terms become operational. We also conduct an ablation study to verify the efficacy of learning multi-modal latent space in our framework and find that learning the functionality latent space directly in one step, i.e., using multi-modal encoders and all loss functions together, does not produce a valid

functionality analysis model without first learning multi-modal latent space. In the following subsection we show the benefits of adopting two-step learning strategy.

6.3. Latent space visualization and discussions

A latent space exploration experiment is designed to illustrate more intuitively how the presented approach establishes the association between multi-modal data through manifold properties. As illustrated in Figure12(a), for two wood material textures that have completely different color and patterns, we first encode them and obtain their coordinates in the texture latent space, then perform linear interpolation between the two coordinates and decode the interpolation results into texture images. Some textures with no practical significance appear in the decoding results, indicating that there is no texture data distribution in the area of the latent space where the interpolation points are located and that it is impossible to determine the semantic relationship between the two textures using the information in the data space alone. The shape latent space, texture latent space, and interaction latent space will be concatenated in the functional latent space, and the dimensionality enhancement provides more channels for correlation establishment. As shown in Figure12(b), after encoding the two textures into the functionality latent space, the linear interpolation between the two coordinates is performed again, and then the interpolation results are decoded into point clouds and reconstructed to IBS. The same interaction relationship can be represented by IBSs with identical topology and analogous shapes. In comparison to the data points at the beginning and end, the middle interpolation result surface has the same Betti number and shape, and both can represent the "supporting" function. In the functionality latent space, the two textures are semantically related, and the data distribution surrounding them indicates that "the part with wood material can be used to support objects."

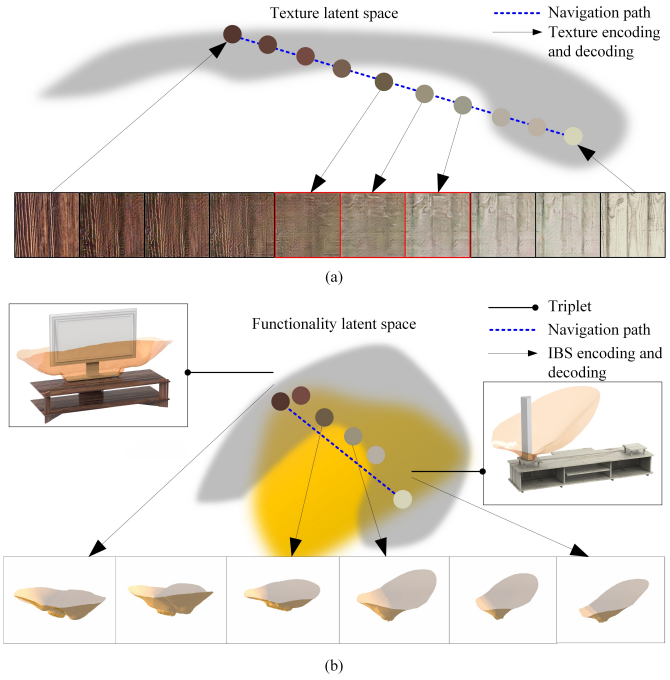

Fig. 12. Navigation in texture latent space (a) and in functionality latent space (b). Two texture images are bridged in functionality latent space through their corresponding interactions and interaction data distribution.

7. Applications

To demonstrate the effectiveness of the functionality latent space, we illustrate its application for functionality-centric shape analysis and synthesis, including functionally-similar shape retrieval, functionality-aware modeling, and functionality-level 3D scene production, all of which are of essence to new graphics techniques and applications.

7.1. Functionally-similar shape retrieval

Similarity-based retrieval of shapes is a common problem in computer graphics. We consider measuring the similarity of two shapes based on their functions. We compare our method to traditional light field shape descriptor (LFD)[46] and ICON[12], which can retrieve shapes with similar functionality from a query interaction scene. Given a query textured shape and its interaction scene, we encode the function cues to latent codes and use the center of the three coordinates as a functionality descriptor. In Figure 13 we visualize distances between the descriptors using t-SNE. LFD considers that the bottles and the TV bench are comparable due to their elongated shapes. ICON distinguishes these shapes based on their interactions and considers the bottles more comparable. Our method, which takes textures into account, indicates that two bottles of Sprite are more similar in functionality. Our method brings the comprehension of shape functionality closer to the semantic level and offers a more thorough description of functionality. Our method is capable of indexing 3D shapes at a higher semantic level, independent of shape geometry and interaction, compared to previous work, as shown in Figure 14.

7.2. Product modeling

We can also use the learned functionality latent space to guide a user in CMF (Colour, Materials, and Finish) design, which is a crucial but tedious process in industrial design. CMF focuses on products' chromatic, tactile, and decorative identity and highly relies on a comprehensive understanding of products' functionality. The automated CMF design process using our method is as follows: based on the input CAD model and the potential interactions of the design, we first calculate the IBS and encode point clouds into the functionality latent space to obtain latent codes; then we retrieve texture images around the latent codes; finally, we add textures to the CAD model parts and render the scene to inspire the CMF design. Figure15 depicts a case study of industrial design product modeling guided by the aforementioned methodology. The algorithm can provide completely different CMF solutions for the same CAD model with different input interactions, indicating that the functionality latent space learned by our method can be used for cross-modal function-related shape attribute editing.

7.3. 3D scene synthesis

As shown in Figure9, we conclude that the learned functionality latent space can be used to synthesize 3D scenes that in-

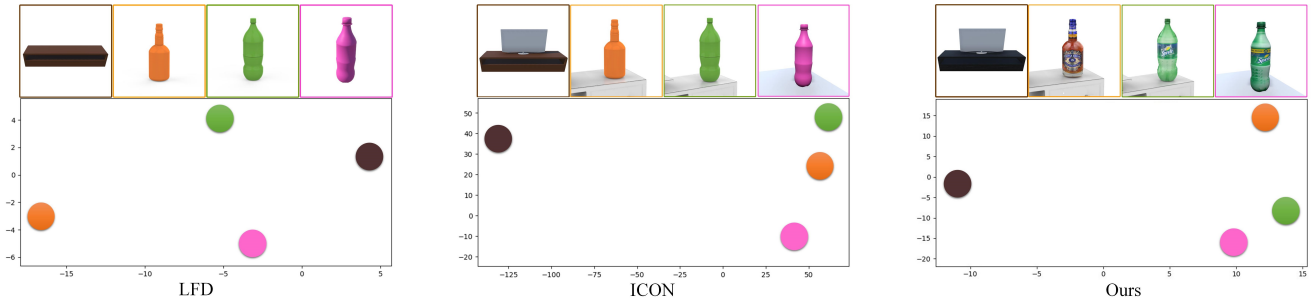

Fig. 13. Two-dimensional t-SNE distance plots between descriptors. Our method can index 3D shapes at a higher semantic level than LFD descriptor and ICON descriptor based on the distances between functionality descriptors.

Fig. 14. Interaction to shape retrieval results by using our method and fSIM-NET [14]. Our method is effective at identifying functional correlations between distinct categories of shapes, yielding more diverse retrieval results that satisfy function requirements.

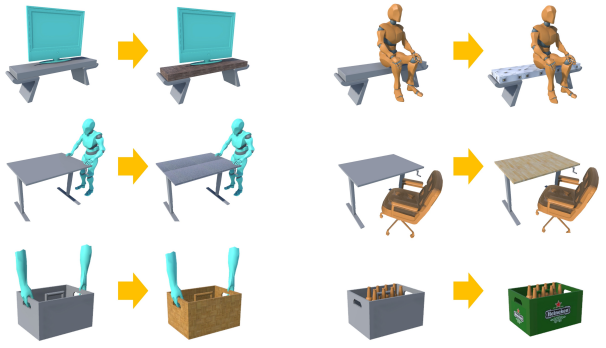

Fig. 15. Product modeling with guidance from the functionality latent space. Given different application scenarios of a CAD model, the functionality latent space can provide diverse texture choices for model parts.

dicate potential uses of 3D objects. This type of scene synthesis can aid in the utilization or simulation of man-made and natural objects, as well as the creation of new products. For content generation in film and game production, designers can easily create 3D scenes by selecting interaction object candidates inferred from functionality latent space, or by allowing the algorithm to add relevant objects, virtual characters, and interactive animations to the target object. This AI design pattern can alleviate the designer's workload. For motion planning and manipulation in machine vision and robotics, a comprehensive understanding of the functionality of 3D objects and the location of subsequent interaction regions is essential.

8. Conclusion and future work

This paper focuses on exploiting cross-domain relationships between function cues and encoding them into a powerful representation of functionality for 3D object parts. We deem a comprehensive understanding of a part's properties necessary for analyzing its functions. Therefore, a single attribute domain cannot define the functionality metric. In addition, there are implicit relationships among properties that cannot be explored or established by traditional clustering's combination. Through multi-modal latent space embedding and interweaving, we discover a latent space of part functionality and obtain a new metric space by scaling part properties to higher dimensions. This allows for the refinement of data distribution and establishes explicit relations between data from various domains. In addition, we conduct a number of experiments to validate the parametric and encoding capabilities of the learned latent space for functionality-centric shape analysis and synthesis.

Limitations. This research has several remaining limitations that suggest immediate generalizations. The current functionality descriptor is based solely on reasoning shapes, textures, and interaction scenes. A deeper understanding of part attributes is required for improved analysis of part functions and the expansion of application scopes. Particularly in the near future, we could investigate more general intra- or inter-object scenarios, such as part mobility analysis and dynamic scene comprehension (involving multi-part interactions and deformations). Second, despite successfully fusing function cues and encoding them into learnable representations, we have not yet provided controllable decoupling of these function cues or part attributes, which could be enhanced with additional loss function design.

Beyond our current technical scope, there are other aspects of data-driven functionality analysis that could be investigated in the long term. Extending our general learning framework to include real-world human behavior, for instance, could shed light on how people typically utilize and interact with an object component. In addition, it would be intriguing to extract spatio-temporal cues from other data modalities, such as videos and RGB-D datasets. These endeavors would enrich and improve all functionality-centric 3D interactive graphics applications.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China No. 62272021, the Research Unit of Virtual Human and Virtual Surgery, Chinese Academy of Medical Sciences under Grant 2019RU004, USA NSF IIS-1715985 and USA NSF IIS-1812606 (awarded to Hong QIN).

References

[1] Funkhouser, T, Min, P, Kazhdan, M, Chen, J, Halderman, A, Dobkin, D, et al. A search engine for 3d models. ACM Transactions on Graphics 2003;22(1):83–105.

[2] Liu, Y, Zha, H, Qin, H. Shape topics: A compact representation and new algorithms for 3d partial shape retrieval. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition; vol. 2. 2006, p. 2025–2032.

[3] Hua, J, Lai, Z, Dong, M, Gu, X, Qin, H. Geodesic distance-weighted shape vector image diffusion. IEEE Transactions on Visualization and Computer Graphics 2008;14(6):1643–1650.

[4] Li, N, Wang, S, Zhong, M, Su, Z, Qin, H. Generalized local-to-global shape feature detection based on graph wavelets. IEEE Transactions on Visualization and Computer Graphics 2015;22(9):2094–2106.

[5] Carlson-Radvansky, LA, Covey, ES, Lattanzi, KM. what effects on where: Functional influences on spatial relations. Psychological Science 1999;10(6):516–521.

[6] Gibson, JJ. The ecological approach to visual perception: classic edition. Psychology Press; 2014.

[7] Hu, R, Savva, M, van Kaick, O. Functionality representations and applications for shape analysis. In: Computer Graphics Forum; vol. 37. 2018, p. 603–624.

[8] Zheng, Y, Cohen-Or, D, Mitra, NJ. Smart variations: Functional substructures for part compatibility. In: Computer Graphics Forum; vol. 32. 2013, p. 195–204.

[9] Zhao, X, Wang, H, Komura, T. Indexing 3d scenes using the interaction bisector surface. ACM Transactions on Graphics 2014;33(3):1–14.

[10] Fish, N, Averkiou, M, Van Kaick, O, Sorkine-Hornung, O, Cohen-Or, D, Mitra, NJ. Meta-representation of shape families. ACM Transactions on Graphics 2014;33(4):1–11.

[11] Hu, R, Li, W, Kaick, OV, Shamir, A, Zhang, H, Huang, H. Learning to predict part mobility from a single static snapshot. ACM Transactions on Graphics 2017;36(6):217:1–217:13.

[12] Hu, R, Zhu, C, van Kaick, O, Liu, L, Shamir, A, Zhang, H. Interaction context (icon) towards a geometric functionality descriptor. ACM Transactions on Graphics 2015;34(4):1–12.

[13] Hu, R, van Kaick, O, Wu, B, Huang, H, Shamir, A, Zhang, H. Learning how objects function via co-analysis of interactions. ACM Transactions on Graphics 2016;35(4):1–13.

[14] Hu, R, Yan, Z, Zhang, J, Van Kaick, O, Shamir, A, Zhang, H, et al. Predictive and generative neural networks for object functionality. ACM Transactions on Graphics 2018;37(4):1–13.

[15] Wang, H, Ho, ES, Shum, HP, Zhu, Z. Spatio-temporal manifold learning for human motions via long-horizon modeling. IEEE Transactions on Visualization and Computer Graphics 2019;27(1):216–227.

[16] Wang, A, Lu, J, Wang, G, Cai, J, Cham, TJ. Multi-modal unsupervised feature learning for rgb-d scene labeling. In: European Conference on Computer Vision. Springer; 2014, p. 453–467.

[17] Huiskes, MJ, Thomee, B, Lew, MS. New trends and ideas in visual concept detection: the mir flickr retrieval evaluation initiative. In: Proceedings of the International Conference on Multimedia Information Retrieval. 2010, p. 527–536.

[18] Guillaumin, M, Verbeek, J, Schmid, C. Multimodal semi-supervised learning for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2010, p. 902–909.

[19] Gong, Y, Wang, L, Hodosh, M, Hockenmaier, J, Lazebnik, S. Improving image-sentence embeddings using large weakly annotated photo collections. In: European Conference on Computer Vision. Springer; 2014, p. 529–545.

[20] Ngiam, J, Khosla, A, Kim, M, Nam, J, Lee, H, Ng, AY. Multimodal deep learning. In: Proceedings of the 28th International Conference on Machine Learning. 2011, p. 689–696.

[21] Srivastava, N, Salakhutdinov, RR. Multimodal learning with deep boltzmann machines. In: Advances in Neural Information Processing Systems. 2012, p. 2222–2230.

[22] Sohn, K, Shang, W, Lee, H. Improved multimodal deep learning with variation of information. In: Advances in Neural Information Processing Systems. 2014, p. 2141–2149.

[23] Tsai, YHH, Huang, LK, Salakhutdinov, R. Learning robust visual-semantic embeddings. In: 2017 IEEE International Conference on Computer Vision. IEEE; 2017, p. 3591–3600.

[24] Kim, T, Cha, M, Kim, H, Lee, JK, Kim, J. Learning to discover cross-domain relations with generative adversarial networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org; 2017, p. 1857–1865.

[25] Pei, Y, Huang, F, Shi, F, Zha, H. Unsupervised image matching based on manifold alignment. IEEE Transactions on Pattern Analysis and Machine Intelligence 2011;34(8):1658–1664.

[26] Laga, H, Mortara, M, Spagnuolo, M. Geometry and context for semantic correspondences and functionality recognition in man-made 3d shapes. ACM Transactions on Graphics 2013;32(5):1–16.

[27] Sun, CY, Zou, QF, Tong, X, Liu, Y. Learning adaptive hierarchical cuboid abstractions of 3d shape collections. ACM Transactions on Graphics 2019;38(6).

[28] Xu, K, Kim, VG, Huang, Q, Kalogerakis, E. Data-driven shape analysis and processing. In: Computer Graphics Forum; vol. 1. 2017, p. 101–132.

[29] Genova, K, Cole, F, Vlasic, D, Sarna, A, Freeman, WT, Funkhouser, T. Learning shape templates with structured implicit functions. In: Proceedings of the IEEE International Conference on Computer Vision. 2019, p. 7154–7164.

[30] Mitra, NJ, Pauly, M, Wand, M, Ceylan, D. Symmetry in 3d geometry: Extraction and applications. In: Computer Graphics Forum; vol. 32. 2013, p. 1–23.

[31] Zheng, Y, Cohen-Or, D, Averkiou, M, Mitra, NJ. Recurring part arrangements in shape collections. In: Computer Graphics Forum; vol. 33. 2014, p. 115–124.

[32] Yumer, ME, Kara, LB. Co-constrained handles for deformation in shape collections. ACM Transactions on Graphics 2014;33(6):1–11.

[33] Xu, C, Chen, Y, Wang, H, Zhu, SC, Zhu, Y, Huang, S. Partafford: Part-level affordance discovery from 3d objects. arXiv preprint arXiv:220213519 2022;.

[34] Kim, VG, Chaudhuri, S, Guibas, L, Funkhouser, T. Shape2pose: Human-centric shape analysis. ACM Transactions on Graphics 2014;33(4):1–12.

[35] Hinkle, L, Olson, E. Predicting object functionality using physical simulations. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE; 2013, p. 2784–2790.

[36] Pirk, S, Krs, V, Hu, K, Rajasekaran, SD, Kang, H, Yoshiyasu, Y, et al. Understanding and exploiting object interaction landscapes. ACM Transactions on Graphics 2017;36(3):1–14.

[37] Zhao, X, Choi, MG, Komura, T. Character-object interaction retrieval using the interaction bisector surface. In: Computer Graphics Forum; vol. 36. 2017, p. 119–129.

[38] Yao, B, Ma, J, Fei-Fei, L. Discovering object functionality. In: Proceedings of the IEEE International Conference on Computer Vision. 2013, p. 2512–2519.

[39] Qi, CR, Su, H, Mo, K, Guibas, LJ. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017, p. 652–660.

[40] Achlioptas, P, Diamanti, O, Mitliagkas, I, Guibas, L. Learning representations and generative models for 3d point clouds. In: International Conference on Machine Learning. 2018, p. 40–49.

[41] Huang, H, He, R, Sun, Z, Tan, T, et al. Introvae: Introspective variational autoencoders for photographic image synthesis. In: Advances in Neural Information Processing Systems. 2018, p. 52–63.

[42] Henzler, P, Mitra, NJ, Ritschel, T. Learning a neural 3d texture space from 2d exemplars. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2020, p. 8356–8364.

[43] Jaderberg, M, Simonyan, K, Zisserman, A, Kavukcuoglu, K. Spatial transformer networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2. 2015, p. 2017–2025.

[44] Massey, WS. A basic course in algebraic topology; vol. 127. Springer; 2019.

[45] Roweis, ST, Saul, LK. Nonlinear dimensionality reduction by locally linear embedding. Science 2000;290(5500):2323–2326.

[46] Chen, DY, Tian, XP, Shen, YT, Ouhyoung, M. On visual similarity based 3d model retrieval. In: Computer Graphics Forum; vol. 22. 2003, p. 223–232.