北京大学 人工智能研究院
INSTITUTE FOR ARTIFICIAL INTELLIGENCE, PEKING UNIVERSITY

PKU-IAI Technical Report: PKU-IAI-2022-T-0002

CoRe Reproducibility Challenge: 3D Semantic Shape Abstraction

Chengying Tu
Yuanpei College
Peking University
2000017802g@stu.pku.edu.cn

Yueru Jia
Yuanpei College
Peking University
2000017824@stu.pku.edu.cn

Abstract

The shape abstraction task is to learn the primitive-based representation of 3D objects. The past work usually learns semantic shape abstraction from mannully labeled part-annotation. Unsupervised learning for 3D semantic shape abstraction remains difficult, because the model cannot understand the affordance of each part and the relationship between parts. In this project, we reproduce two off-the-shelf well-performing shape abstract models, Cuboid Shape Abstraction via Joint Segmentation and Neural Parts, on the PartNet dataset, and analyze their respective task perspectives and methodical characteristics. The former method uses the traditional cuboid primitive and tries to map the point cloud to a compact cuboid representation. And the latter proposes a new 3D primitive representation, which realizes homomorphic mapping between the sphere and the target object, and is more flexible than the traditional primitive representation. Our intuition is that Neural Parts will perform better on semantic shape abstraction. We test our intuition experimentally, but only in the chair category. Based on the experimental results, we analyze the advantages and disadvantages of the two.

1 Introduction

Human-made objects tend to have delicate structures and can be decomposed into simpler, more regular parts. The semantic information contained in the structure is formed by the relationships among the decomposed parts. It provides a way to understand the functionality or affordance of an object at the part level. For example, the common chair can be decomposed into the cushion, the backrest, and the legs. They interact with each other to form the overall affordance and each part has its functionality.

Based on the idea above, we focus on shape abstraction tasks. This task is to learn a structured representation of 3D objects, that is, how to decompose complex 3D objects into simple and meaningful geometric primitives. The past visual tasks tended to focus only on geometric information, while semantic information was manually added for the model to learn [16]. That is because humans can easily obtain the function of each part and the relationship between each part from visual geometry information. However, learning structured representations of 3D objects containing semantic information in an unsupervised way remains challenging.

Unsupervised shape abstraction integrates 3D shapes through geometric primitives while maintaining a consistent structure in a collection of shapes. In this project, we focus on two unsupervised 3D shape abstraction methods: Cuboid Shape Abstraction via Joint Segmentation [15] and Neural Parts [9].

The former is the state-of-art method of 3d shape abstraction based on cuboid primitive and the latter proposes a new flexible 3D primitive representation method to capture arbitrarily complex geometric figures. We hope to take these two advanced shape abstraction methods as examples to compare and analyze the influence of traditional primitives with fixed shapes (such as cuboids) and primitive representations with flexible shapes on semantic information learning when performing 3D shape abstraction tasks, especially in datasets containing fine-grained semantic information. The visual difference between these two methods is shown in Figure 1. This is very important for application scenarios such as 3D shape reconstruction and structured shape generation.

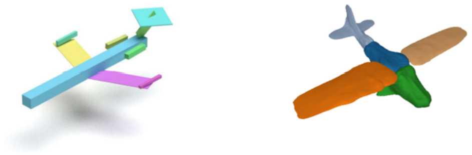

Figure 1: Examples that show different primitive representations for the two methods. The left is generated by Cuboid Via Joint Segmentation and the right by Neural Parts.

We select PartNet [6], a dataset containing fine-grained semantic annotations based on ShapeNet [2], as our experimental dataset. The structured representation of each object category requires separate training of corresponding models. Due to limited time, we choose chairs as our experimental object category. By comparing the semantic annotation generated by the model through unsupervised learning with PartNet's ground-truth, we generate quantitative results by calculating mIoU and Chamfer Distance(CD). We also try to visualize the results of the two methods to analyze their respective characteristics.

2 Related Work

2.1 Shape Abstraction

Methods of shape abstraction are aimed at decomposing complex 3D objects into semantically meaningful simpler geometric primitives. Traditional methods tend to use supervision learning to learn the primitive-based representation [7, 11, 17], namely the primitive parameters. Traditional primitives include cuboids [14], superquadrics [8], convexes [4], CSG trees [10] and shape programms [13]. Cuboid Via Joint Segmentation [15] we reproduced in this project uses cuboid primitive but is trained without any part-level annotations, namely unsupervised learning.

Recent studies on 3D reconstruction focus as well on 3D shape abstraction without any part-level annotations [8, 14, 9]. Among them, Neural Parts [9] is a primitive-based method with great performance. Neural Parts proposes a new primitive representation to represent parts more flexibly. The primitive contains not only the geometric features of the shape components, but also semantic information between parts.

2.2 Semantic Analysis of Shape

Semantic analysis of 3D shapes is an important problem in computer vision. In the absence of a large-scale annotated dataset, early research work could only qualitatively evaluate the results of algorithms. Attene et al. [1] qualitatively use 11 3D surface meshed to compare five mesh segmentation algorithms. Collecting 380 surface meshes from 19 object classes, Chen et al. [3] performs instance-level partial decomposition of each shape, and proposes quantitative evaluation metrics for shape abstraction. Hu et al. [5] tries to study co-segmentation between similar shapes.

In recent years, the development of deep learning has promoted the birth of large annotated datasets for shape abstraction. Although the two unsupervised learning methods we reproduced in this project do not directly learn from labels, large annotated datasets can help test the performance of the models. Chang et al. [2] collects large-scale synthetic 3D CAD models as ShapeNet and includes over 3 million models and 3135 object classes. Furthermore, the ShapeNet models are annotated by an active learning approach [16]. And Mo et al. [6] provides more fine-grained part annotations as PartNet with each shape containing an average of 18 parts.

3 Method

The two methods we reproduced focus on different task pointcuts. The task of Cuboid Via Joint Segmentation is to assign a parameterized cuboid representation to the input 3D point cloud and segment the point cloud accordingly. And Neural Parts focuses on how to reconstruct the 3D object in a structured representation from a 2D image. Despite their different drives, they both learned how to abstract shapes, namely the structured representation of objects. In this section, we will briefly introduce the two methods and their characteristics to help subsequent experimental analysis.

3.1 Cuboid Via Joint Segmentation (Unsupervised learning for cuboid shape abstraction via joint segmentation from point clouds)

3.1.1 Introduction

Yang and Chen [15] proposes the joint tasks of segmentation and shape abstraction to assign points to different cuboid primitives, which can extract a more common structure of the whole dataset to avoid ambiguity and degeneration.

By adding the segmentation branch, Yang and Chen [15] is able to extract finer structures of objects even without supervised information. Figure 2. of segmentation abstraction results with different point numbers shows that semantic and structural information can be obtained by even fewer points. It can also be extended to a range of applications, like shape generation and shape interpolation.

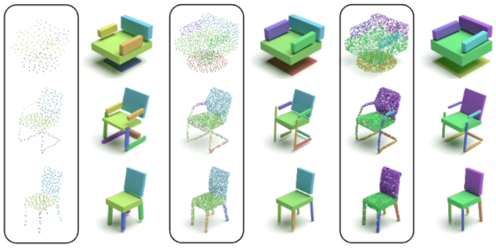

Figure 2: Segmentation and abstraction results of Cuboid Via Joint Segmentation trained with different point numbers.

3.1.2 Details

Given a point cloud *P* with N points, Yang and Chen [15] uses an unsupervised shape abstraction method to map it into a fixed M-cuboid representation. Each cuboid ($p_m = [t_m; r_m; s_m; \theta_m]$) is described by four parameters, cuboid-related translation, rotation, scale, and existence, which will be predicted by the network.

The general structure is a variational auto-encoder(VAE) consisting of two stages: encoding and decoding and contains three parts, feature embedding network, shape abstraction network, and cuboid-associated segmentation network.

First, the feature embedding network extracts N point-wise features f^p to obtain the global feature f^g, then map it into a latent code z. To train the whole network in an unsupervised method, z goes through a two-branch sub-network to decode the parameters of M cuboids. The shape abstraction network decodes z to obtain the required parameters. The segmentation branch performs M-label point cloud segmentation using the attention module. The model is trained using a novel loss with four components, the reconstruction loss L_{recon} to ensure the consistency between the part segmentation and reconstructed cuboids, the compactness loss $L_{compact}$ to encourage a more compact representation, the existence loss L_{exist} to predict the existence of each cuboid, the KL-divergence loss L_{KL-div} for distribution constraints.

3.2 Neural Parts

3.2.1 Introduction

Paschalidou et al. [9] proposes a new 3D primitive representation, which does not predict primitive parameters directly like traditional methods, but uses Invertible Neural Network (INN) to define primitives, and realizes homomorphic mapping between spheres and target objects.

Because of the limited expressiveness of traditional primitives, the greater the number of primitives, the higher the precision of reconstruction. Different from that, the new primitive proposed by Neural Parts is more flexible and can capture arbitrarily complex geometric shapes. In such cases, specifying different numbers of primitives will have different expressive effects (shown in Figure 3). If the number of primitives is set well, the resulting primitive representation can contain very full and precise semantic information.

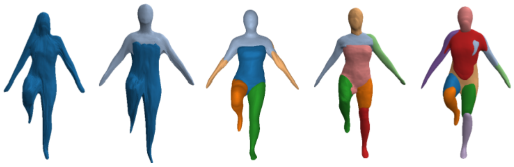

Figure 3: The primitives predicted by Neural Parts using 1, 2, 5, 8, and 10 primitives.

3.2.2 Details

Given an input image, Paschalidou et al. [9] seeks to learn a representation with M primitives that best describes the target object. The primitives here are defined via a deformation between shapes that is parametrized as a learned homeomorphism implemented with an Invertible Neural Network (INN). Specifically, for each primitive, a homeomorphism between the 3D space of a simple genus-zero sphere shape (the latent space) and the 3D space of the target object (the primitive space) will be learned. So the deformed shape will match a part of the target object.

The model comprises two main components: a feature extractor and a conditional homeomorphism.

First, the feature encoder maps the input to a global feature representation $\mathbf{F}$. Then, for every primitive m, $\mathbf{F}$ is concatenated with a learnable primitive embedding $\mathbf{P}_m$ to generate the shape embedding $\mathbf{C}_m$ for this primitive. The Conditional Homeomorphism $\phi_\theta(\cdot; \mathbf{C}_m)$ is implemented by a stack of L conditional coupling layers. Applying the forward mapping on a set of points $\mathcal{Y}_s$, randomly sampled on the surface of the sphere, generates points on the surface of the m-th primitive $\mathcal{X}_p^m$. Using the inverse mapping $\phi_\theta^{-1}(\cdot; \mathbf{C}_m)$ can help compute whether any point in 3D space lies inside or outside a primitive. The model is trained using both $surface(\mathcal{L}_{rec}, \mathcal{L}_{norm})$ and $occupancy(\mathcal{L}_{occ})$ losses to simultaneously capture fine object details and volumetric characteristics of the target object. The use of inverse mapping can help impose additional constraints (e.g. discouraging inter-penetration) on the predicted primitives $(\mathcal{L}_{overlap}, \mathcal{L}_{cover})$.

4 Experiments and Analysis

4.1 Dataset

We implement these two works on the chair category of the PartNet dataset [6]. For the chair category annotations, the number of shape annotations, the number of distinct shape instances, and the number of shapes are respectively 8176, 6400, and 77. The number of different part semantics and part instances finally collected are 57 and 176k.

4.2 Experimental Setup

For Cuboid Via Joint Segmentation, we adapt the settings and hyper-parameters from the original paper for training. And we set M to 16 for the fixed M-cuboid representation and pre-align and normalize the shapes to the unit scale. As for Neural Parts, since the number of primitives should be manually set up, we select 5 for better performance on the chair category. For further experiments on chairs with more complex structures or other categories, the number should be adjusted accordingly.

We make modifications as the PartNet suggests that the input point size is changed to 10,000 points, and all input shapes are guaranteed to have 10,000 points.

4.3 Qualitative Analysis

We visualized the predicted primitives of the chair category.

The visualization of Cuboid Via Joint Segmentation is shown in Figure 4. From the visualization we can see that the cuboid primitive representation can capture the general structure and the shape of components. But in terms of fine grain, for example, the connecting parts of different components are blank. The representation works best for chairs square in shape, as in the last example. However, for chairs with more curved surfaces, such as the second example, cuboid representation will be messy and unable to better abstract shapes. Unlike the Neural Parts method, this method has only an upper limit on the number of cuboids, so the result may have fewer cuboid primitives than the set number, as in the first example. A smaller number of primitives can sometimes achieve better shape abstraction results.

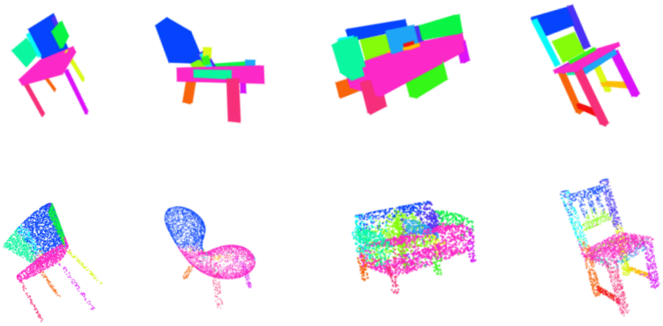

Figure 4: The visualization of the shape abstraction and segmentation tasks in Cuboid Via Joint Segmentation for the chair category.

The visualization of Neural Parts is shown in Figure 5. We can see that shape abstraction results of Neural Parts are highly consistent for the same category. And we can also see the uniqueness of the primitive representation of Neural Parts. Compared with the cuboid primitive, the primitive representation of Neural Parts is more consistent with the original object shape. As mentioned in the previous method introduction, if the number of primitives is changed, the abstracted shape of the part may go different.

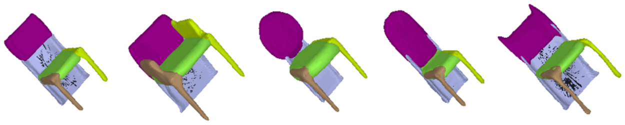

Figure 5: The visualization of Neural Parts with 5 primitives for the chair category. The black points indicate unlabeled points.

4.4 Quantitative Analysis

We are going to use the mean Intersection-over-Union (mIoU) scores and Chamfer Distance(CD) [12] as our evaluation metric. As for mIoU, we first remove the unlabeled ground-truth points, and then for each object category, calculate the IoU between the predicted point set and the ground-truth point set for each semantic part category across all test shapes. For each object category, the mIoU is the average of the per-part-category IoUs. Lager mIoU means better performance. And Chamfer Distance defines the distance between two point clouds. It finds the nearest point in the other point set for each point in each cloud, and sums the square of distance up. It is often used for 3D shape reconstruction task. Smaller Chamfer Distance means better performance.

The experimental results are shown in Table 1. We can see that overall the results of Neural Parts outperforms Cuboid Via Joint Segmentation.

For mIoU metric, the performance of the two methods is similar, showing that the two methods works both reasonably for the shape abstraction of each part. The reason why Neural Parts perform slightly

Table 1: We compare Cuboid Via Joint Segmentation with 16 cuboids and Neural Parts with 5 primitives.

	mIoU	Chamfer Distance
Cuboid Via Joint Segmentation	0.803	0.435
Neural Parts with 5 primitives	0.825	0.025

better than Cuboid Via Joint Segmentation might be because the primitive expression effect of Neural Parts is naturally more delicate than cuboid in detail processing.

For chamfer distance metric, the performance of Neural Parts is far better than that of Cuboid Via Joint Segmentation, though possibly influenced by different point cloud sampling methods in calculating L2 CD, as well as the instability of results caused by different settings of the primitive number. On in-depth study of the network structure and loss function of the two, we find that since the shape of Neural Part's primitive defined by the Invertible Neural Network (INN) is flexible, special loss function - occupancy loss and surface loss - are designed to prevent the shape range of a component from being too large or too small. For example, a primitive represents four chair legs, which should be punished. Although Cuboid Via Joint Segmentation also designs compactness loss to punish redundant cuboids, it still chooses to add redundant cuboids for representation when there are many curved surfaces in the shape of objects.

5 Discussion

The difficulties of our project are setting up the environment and loading the large dataset. The task focus of the two methods we reproduce are different. Cuboid Via Joint Segmentation focuses on the mapping from point cloud to cuboid primitive and the corresponding semantic segmentation of point cloud, while Neural Parts focus on 3D reconstruction. Therefore, the input of the two methods is actually different. The former is a 3D point cloud, while the latter is a 2D image. But both ultimately learn the primitive-based representation of shape abstraction.

Experiments are conducted to compare the difference between the shape abstract results predicted by the two methods and the ground truth. We choose mIoU and Chamfer Distance(CD) as two metrics for evaluation. Experimental results show that Neural Parts can be more flexibly adapted to semantic shape abstraction tasks by implicitly defining the primitive with Invertible Neural Network (INN). The possible disadvantage is that the Neural Parts need to specify the number of primitives beforehand, which is fixed for training and further testing. We know that different objects have different numbers of parts. So it may influence the performance of Neural Parts on complicated 3D objects. However, Cuboid Via Joint Segmentation only has an upper limit on the number of primitives, which means that it tolerates redundant primitives. For a primitive, it can choose not to express it. Thus, although Neural Parts outperforms the Cuboid Via Joint Segmentation in the chair category, it is uncertain in the wider category of objects.

Each 3D object category needs to be trained on a separate model. Due to time and computational constraints, we spent a lot of time on environment setup and code modification, so we only conducted experiments on the chair category. The analysis of our experimental results is also limited to the category of chairs, which is a pity for us.

For future work, we can do more ablation experiments to study the influence of different settings of coefficient parameters, and the number of input point clouds. We also plan to evaluate the two models on more object categories.

Author Contribution

In this course project, Chengying reproduced the model Neural Parts on PartNet and wrote this report. Yueru reproduced the model Cuboid Via Joint Segmentation on PartNet and did the representation.

References

[1] Marco Attene, Sagi Katz, Michela Mortara, Giuseppe Patané, Michela Spagnuolo, and Ayellet Tal. Mesh segmentation-a comparative study. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 7–7. IEEE, 2006. 2

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2

[3] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. A benchmark for 3d mesh segmentation. *Acm transactions on graphics (tog)*, 28(3):1–12, 2009. 2

[4] Matheus Gadelha, Giorgio Gori, Duygu Ceylan, Radomir Mech, Nathan Carr, Tamy Boubekeur, Rui Wang, and Subhransu Maji. Learning generative models of shape handles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 402–411, 2020. 2

[5] Ruizhen Hu, Lubin Fan, and Ligang Liu. Co-segmentation of 3d shapes via subspace clustering. In *Computer graphics forum*, volume 31, pages 1703–1713. Wiley Online Library, 2012. 2

[6] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019. 2, 4

[7] Chengjie Niu, Jun Li, and Kai Xu. Im2struct: Recovering 3d shape structure from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4521–4529, 2018. 2

[8] Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1060–1070, 2020. 2

[9] Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, and Sanja Fidler. Neural parts: Learning expressive 3d shape abstractions with invertible neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3204–3215, 2021. 1, 2, 4

[10] Gopal Sharma, Rishabh Goyal, Difan Liu, Evangelos Kalogerakis, and Subhransu Maji. Csgnet: Neural shape parser for constructive solid geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5515–5523, 2018. 2

[11] Gopal Sharma, Difan Liu, Subhransu Maji, Evangelos Kalogerakis, Siddhartha Chaudhuri, and Radomír Měch. Parsenet: A parametric surface fitting network for 3d point clouds. In *European Conference on Computer Vision*, pages 261–276. Springer, 2020. 2

[12] Dmitriy Smirnov, Matthew Fisher, Vladimir G Kim, Richard Zhang, and Justin Solomon. Deep parametric shape predictions using distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 561–570, 2020. 5

[13] Yonglong Tian, Andrew Luo, Xingyuan Sun, Kevin Ellis, William T Freeman, Joshua B Tenenbaum, and Jiajun Wu. Learning to infer and execute 3d shape programs. *arXiv preprint arXiv:1901.02875*, 2019. 2

[14] Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2635–2643, 2017. 2

[15] Kaizhi Yang and Xuejin Chen. Unsupervised learning for cuboid shape abstraction via joint segmentation from point clouds. *ACM Transactions on Graphics (TOG)*, 40(4):1–11, 2021. 1, 2, 3

[16] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 1, 2

[17] Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 900–909, 2017. 2