Computational Visual Media
DOI 10.1007/s41095-022-0281-9 Vol. 9, No. x, month year, xx–xx
Research Article

Semi-supervised 3D shape segmentation with multilevel consistency and part substitution

Chun-Yu Sun[1], Yu-Qi Yang[1], Hao-Xiang Guo[1], Peng-Shuai Wang[2], Xin Tong[2], Yang Liu[2]✉, and Heung-Yeung Shum[1]

Abstract The lack of fine-grained 3D shape segmentation data is the main obstacle to developing learning-based 3D segmentation techniques. We propose an effective semi-supervised method for learning 3D segmentations from a few labeled 3D shapes and a large amount of unlabeled 3D data. For the unlabeled data, we present a novel *multilevel consistency* loss to enforce consistency of network predictions between perturbed copies of a 3D shape at multiple levels: point-level, part-level, and hierarchical level. For the labeled data, we develop a simple yet effective part substitution scheme to augment the labeled 3D shapes with more structural variations to enhance training. Our method has been extensively validated on the task of 3D object semantic segmentation on PartNet and ShapeNetPart, and indoor scene semantic segmentation on ScanNet. It exhibits superior performance to existing semi-supervised and unsupervised pre-training 3D approaches.

Keywords shape segmentation, semi-supervised learning, multilevel consistency.

1 Introduction

Recognizing semantic parts of man-made 3D shapes is an essential task in computer vision and graphics. Man-made shapes often consist of fine-grained and semantic parts, many of which are small and hard to distinguish. Moreover, for 3D shapes within a shape category, the existence, geometry, and layout of semantic parts can often have large variations. As a result, obtaining accurate and consistent fine-grained segmentation for a shape category is challenging, even for human workers.

Recently, supervised learning approaches have been widely used in shape segmentation; they need sufficient labeled data. However, as there are not many large well-annotated 3D datasets, and the 3D data labeling process is costly and tedious, it is difficult to apply these methods to shape categories with limited labeled data. In this paper, we propose a novel semi-supervised approach for fine-grained 3D shape segmentation. Our method learns a deep neural network from a small set of segmented 3D point clouds and a large number of unlabeled 3D point clouds within a shape category, thus greatly reducing the workload of 3D data labeling.

We propose two novel schemes to efficiently utilize both unlabeled and labeled data for network training. For unlabeled data, inspired by the pixel-level consistency scheme used in semi-supervised image segmentation [30, 40], we propose a set of *multilevel consistency* losses for measuring the consistency of network predictions between two perturbed copies of a 3D point cloud at the *point-level*, *part-level*, and *hierarchical level*. Via the multilevel consistency, the data priors hidden in the unlabeled data can be learned by the network to good effect. For the available labeled shapes, we present a simple yet effective multilevel *part-substitution* algorithm to enrich the labeled data set by replacing parts with semantically similar parts of other labeled data. The algorithm is specially designed for 3D structured shapes, like chairs and tables, and it enhances the geometry and structural variation of the labeled data in a simple way, leading to a boost in network performance.

We evaluate the efficacy of our method on the task of 3D shape segmentation including object segmentation and indoor scene segmentation, by training the networks with different amounts of

1 C-Y. Sun, Y-Q. Yang, H-X. Guo and H-Y. Shum are with Institute for Advanced Study, Tsinghua University.
2 P-S. Wang, X. Tong and Y. Liu are with Microsoft Research Asia.

labeled data and unlabeled data. An ablation study further validates the significance of each type of consistency loss. Extensive experiments demonstrate the superiority of our method over state-of-the-art semi-supervised and unsupervised 3D pretraining approaches. Our code and trained models are publicly available at `https://isunchy.github.io/projects/semi_supervised_3d_segmentation.html`.

2 Related Work

In this section, we briefly review related 3D shape segmentation approaches and shape synthesis techniques.

2.1 Unsupervised 3D segmentation

Early attempts at unsupervised segmentation focused on decomposing a single shape into meaningful geometric parts using clustering, graph cuts, or primitive fitting (see surveys in [45, 46]). To obtain consistent segmentation within a shape category, a series of unsupervised co-segmentation works (see surveys in [45, 67]) proposed exploiting geometrically similar parts across over-segmented shapes, via feature co-analysis or co-clustering. Learning a set of primitives to represent shape is another approach to shape decomposition and segmentation, e.g. using cuboids [50, 54], superquadrics [41], convex polyhedra [13], or implicit functions [19]. Chen *et al.* [9] trained a branched autoencoder network, Bae-Net for shape segmentation, in which each branch learns an implicit representation for a meaningful shape part. All the above methods rely on geometric features for segmentation and do not take semantic information into consideration, which may lead to results inconsistent with human-defined semantics.

2.2 Supervised 3D segmentation

Various supervised methods perform 3D segmentation using deep neural networks trained on a large number of labeled 3D shapes or scenes [22]. Xie *et al.* [66] project a 3D shape into multiview images and use 2D CNNs to enhance the segmentation. Kalogerakis *et al.* [28] combine CRF with multiview images to boost segmentation performance. Dai *et al.* [12] back-project the feature learned by multiview images to 3D to conduct scene segmentation. Qi *et al.* [43] use a point-based network to predict per-point semantic labels by combining global and pointwise features. Other works [35, 44, 53] enhance feature propagation by using per-point local information. Wang *et al.* [62] and Hanocka *et al.* [23] build graphs from point sets and conduct message passing on graph edges while further methods [29, 36, 42, 70] directly perform CNN computation on mesh surfaces. Song *et al.* [49] conduct scene semantic segmentation with the help of the scene completion task. For efficiency, many works [10, 20, 27, 58, 73] use sparse voxels or supervoxels to reduce the computational and memory costs while achieving better segmentation results. Unlike these supervised methods that require a large amount labeled data, we leverage a few labeled data items and a large amount of unlabeled data for effective segmentation.

2.3 Weakly-supervised 3D segmentation

3D shapes in many shape repositories are modeled by artists and often come with rich metadata, like part annotations and part hierarchies. Although part-related information may be inconsistent with the ground truth, it can be used to weakly supervise the training of shape segmentation networks. Yi *et al.* [71] learn hierarchical shape segmentations and labeling from noisy scene graphs from online shape repositories and transfer the learned knowledge to new geometry. Muralikrishnan *et al.* [39] discover semantic regions from shape tags. Wang *et al.* [61] learn to group existing fine-grained and meaningful shape segments into semantic parts. Sharma *et al.* [47] embed 3D points into a feature space based on the annotated part tag and group hierarchy and then fine-tune the point features with a few labeled 3D data items for shape segmentation. Zhu *et al.* [76] utilize part information from a 3D repository to train a part prior network for proposing per-shape parts for an unsegmented shape, then train a co-segmentation network to optimize part labelings across the input dataset. Xu *et al.* [69] learn shape segmentation with an assumption that each shape in the large training dataset has at least one labeled point per semantic part. Unlike these weakly supervised methods, our method requires no additional weak supervision on unlabeled data.

2.4 Unsupervised 3D pretraining

Unsupervised pretraining [4] has demonstrated its advantage in many computer vision and natural language processing tasks, where a feature encoding network is pretrained on a large amount of unlabeled data and then is fine-tuned for downstream tasks using a small amount of labeled data. For 3D analysis tasks, Hassani and Haley [24] pretrain a multi-scale graph-based encoder with the ShapeNet dataset [6] using a multi-task loss. Wang *et al.* [60] use multiresolution instance discrimination loss for pre-training, while Hou *et al.* [25] and Xie *et al.* [64] employ contrastive loss.

Instead of using this two-step training: pretraining and fine-tuning, our network is trained with both labeled and unlabeled data from the beginning. Given the same amount of labeled data, our semi-supervised method is superior to a fine-tuned pretrained network on 3D object segmentation and indoor scene segmentation.

2.5 Semi-supervised segmentation

Semi-supervised learning tries to employ unlabeled data to facilitate supervised learning, thus reducing the amount of labeled data needed for training; see [55] for a detailed survey. Many approaches were first developed for image classification, like temporal ensembling [31] that aggregates the prediction of multiple previous network evaluations, Mean-Teacher [51] that averages model weights instead of predictions, FixMatch [48] that uses confidence-aware pseudo-labels of weakly-augmented data to guide the strongly-augmented data prediction, and MixMatch [5] that guesses low-entropy labels for data-augmented unlabeled data and mixes labeled and unlabeled data using MixUp. For image segmentation, Ouali *et al.* [40] utilizes *cross-consistency* to train image segmentation networks, where pixel features extracted by the encoder are perturbed and enforced to be consistent with network predictions after decoding. Ke *et al.* [30] use two networks with different initializations and dynamically penalize inconsistent pixel-wise predictions for the same image input. French *et al.* [15] improve image segmentation accuracy by imposing strong augmentation on unlabeled training images via region masking and replacement. Wang *et al.* [56] employ the Mean-Teacher model with improved uncertainty computation and use auxiliary tasks with task-level consistency for medical image segmentation. Unlike the above semi-supervised image segmentation methods that leverage pixel-level consistency only, or use task-level consistency, our approach utilizes 3D shape part hierarchy and maximizes 3D segmentation consistency at multiple levels, including point-level, part-level and hierarchical level.

For 3D segmentation, Bae-Net [9] can learn a branched network from labeled data and unlabeled data for shape segmentation. Although this approach works well for segmenting 3D shapes into a few large parts, it is nontrivial to extend it to many fine-grained semantic 3D segments due to its large network size, and it is unclear whether it can handle the large variety of part structures well. Wang *et al.* [57] propose to retrieve a similar 3D shape with part annotations from a mini-pool of shape templates for a given input 3D shape, and learn a transformation to morph the template shape towards the input shape. From the transformed template, a part-specific probability space is learned to predict point part labels, and part consistency within the training batch is utilized. However, its prediction accuracy can be severely affected by the chosen template and the deformation quality.

2.6 Structure-aware shape synthesis

A set of geometric operations has been developed for generating 3D shapes from shape parts, such as part assembly [7, 17, 65], structural blending [2], and set evolution [68]. Although these methods are effective for generating high-quality 3D shapes, some require special pre-processing and interactive editing. Recent methods composite [26, 63, 77] or edit shapes by learning structural variations within a large set of segmented 3D shapes [37]. Another set of methods [16, 21, 75] utilizes the functionality of shape structures to guide 3D shape synthesis. In our work, we develop a simple and automatic part substitution scheme for generating shapes with proper structural and geometric variations from a small number of labeled 3D shapes, whose quality is sufficient to improve network training. We also notice that recent point cloud augmentation techniques [8, 32, 34] that mix points of different shapes randomly to generate more varied shapes can enhance point cloud classification, and can be extended to shape segmentation [72]. However, random augmentation does not respect shape structure and can lead to limited improvements only, as our experiments show. Instead, our part substitution scheme enriches structural variations of the labeled dataset and improves the network performance.

3 Method Overview

3.1 Input and output

We assume a set of fine-grained 3D semantic part labels probably with a structural hierarchy is pre-defined for a 3D shape category. For instance, at a coarse level the structure of a chair includes the back, the seat, and the support; the chair support can be decomposed at a finer level into vertical legs, horizontal supports, and other small parts. We denote the number of hierarchical levels by K; the K-th level is the finest level.

Our goal is to predict hierarchical part labels for each point of the input point cloud and hence determine its shape part structure. The training data includes a small set of labeled point clouds and a large number of unlabeled point clouds. All point clouds are sampled from shapes within the same shape category, so their part structures are implicitly coherent but nevertheless

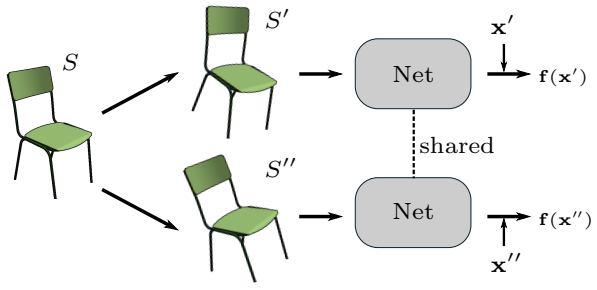

Fig. 1 General neural network setup for 3D semantic segmentation. The network takes a point cloud S as input and feeds two perturbed copies of S: S' and S'', to the network, separately. The output point features $\mathbf{f}(\mathbf{x})$ of point $\mathbf{x}$ are transformed to probability vectors $\mathbf{p}^{(k)}(\mathbf{x}_i)$ for determining the segmentation part label at the k-th level. Multilevel consistency is built upon the probability vectors of points of S' and S''.

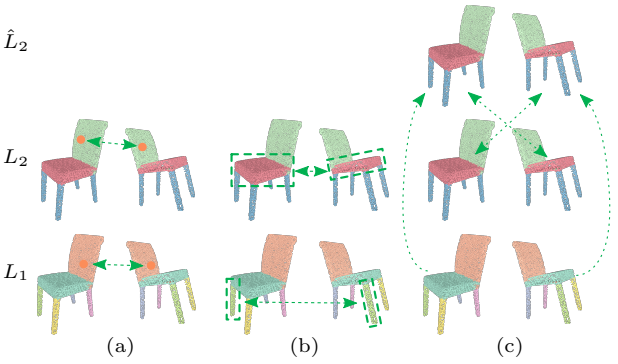

Fig. 2 Multilevel consistency on two perturbed copies of a chair model having a 2-level hierarchy. L_1 and L_2 are the fine and coarse levels, respectively. Point color corresponds to predicted part label at each level. The segmentation of $\hat{L}_2$ is the pseudo-part prediction at L_2 inferred from L_1 according to the predefined shape hierarchy. (a) Point-level consistency built on the corresponding point pairs between two copies at each level. (b) Part-level consistency built on parts with the same semantics between two copies at each level. (c) hierarchical consistency built on the corresponding points between the shape copies on $\hat{L}_2$ and L_2.

have topological and geometric variations.

3.2 Base network

Our semi-supervised learning relies on a 3D network that takes a 3D point cloud as input and outputs the point features. Each point feature is transformed to probability vectors via two fully-connected (FC) layers and a softmax function for determining the segmentation labels at each granularity level. We defer the exact choice of our network structure to Section 6.

3.3 Data perturbation for semi-supervised training

For an input point cloud S which is scaled uniformly to fit within a unit sphere, we generate two randomly-perturbed copies of S, denoted S' and S'', and pass them to the network during the training stage. In our implementation, the perturbation is composed of a uniform scaling within the interval $[0.75, 1.25]$, a random rotation whose pitch, yaw, and roll rotation angles are less than 10°, and random translations along each coordinate axis within the interval $[-0.25, 0.25]$. The perturbed point cloud is clipped by the unit box before input to the network. This perturbation strategy follows the approach of [60] for unsupervised pre-training. Data perturbation makes the trained network more robust and helps build our multilevel consistency between the perturbed shape copies. $\mathbf{x}'_i$ and $\mathbf{x}''_i$ are the perturbed copies in S' and S'' respectively of $\mathbf{x}_i$ in S. The network with perturbed data is illustrated in Fig. 1.

3.4 Notation

We use the following notation in the paper:

S : the input point cloud, $\{\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^3\}$.

$L^{(k)} \in \mathbb{N}^+$: The number of semantic labels at the k-th level.

$\mathbf{p}^{(k)}(\mathbf{x}_i) \in \mathbb{R}^{L^{(k)}}$: probability vector of $\mathbf{x}_i$ at the k-th level.

$\mathbf{q}^{(k)}(\mathbf{x}_i) \in \mathbb{R}^{L^{(k)}}$: one-hot vector for $\mathbf{x}_i$ at the k-th level, corresponding to the ground-truth semantic label of $\mathbf{x}_i$.

3.5 Loss design

For the labeled point cloud S, we use the cross-entropy loss to penalize dissimilarity of point semantic labels of S' and S'' to the ground truth labels at multiple levels, as follows:

$$L_{\text{seg}}(S', S'') = \frac{1}{2n} \sum_{i=1}^{n} \sum_{k=1}^{K} \Big[g_{\text{ce}} \left(\mathbf{q}^{(k)}(\mathbf{x}_i), \mathbf{p}^{(k)}(\mathbf{x}'_i) \right) + g_{\text{ce}} \left(\mathbf{q}^{(k)}(\mathbf{x}_i), \mathbf{p}^{(k)}(\mathbf{x}''_i) \right) \Big], \tag{1}$$

where $g_{\text{ce}}(\cdot, \cdot)$ is the standard cross-entropy loss.

For both unlabeled and labeled inputs, we use the multilevel consistency loss introduced in Section 4 to ensure the network outputs of S' and S'' are consistent with each other.

3.6 Labeled data augmentation

The structure of labeled 3D shapes offers a great possibility for synthesizing new shapes with semantics. In Section 5, we propose a simple part-substitution method to enrich the labeled shape set, which can improve the performance of both supervised and semi-supervised approaches.

4 Multilevel consistency

We now introduce our multilevel consistency for utilizing unlabeled data for network training. The multilevel consistency builds on point-level (Section 4.1), part-level (Section 4.2), and hierarchical level (Section 4.3) consistency, and is illustrated in Fig. 2.

4.1 Point-level consistency

A pair of points, $\mathbf{x}'_i \in S'$ and $\mathbf{x}''_i \in S''$, should have probability vectors as similar as possible due to self-consistency (see Fig. 2(a)). Based on this property, we build a point-level consistency loss L_{point} upon their probability vectors using the symmetric KL-divergence loss D_{KL}:

$$L_{\text{point}} = \frac{1}{2n}\sum_{i=1}^{n}\sum_{k=1}^{K}\Big[D_{KL}\left(\mathbf{p}^{(k)}(\mathbf{x}'_i) \parallel \mathbf{p}^{(k)}(\mathbf{x}''_i)\right) + D_{KL}\left(\mathbf{p}^{(k)}(\mathbf{x}''_i) \parallel \mathbf{p}^{(k)}(\mathbf{x}'_i)\right)\Big]. \tag{2}$$

Point-level consistency is a simple extension of pixel-level consistency which has been extensively used in semi-supervised image segmentation. The KL-divergence loss can be replaced with the MSE loss; the latter has a better performance on semi-supervised image classification in [31, 51]. However, we found that they have similar performance on 3D segmentation, and indeed the former is slightly better.

4.2 Part-level consistency

Due to data perturbation, the predicted part distributions of S' and S'' at the same part level can be different. We impose a novel part-level consistency to minimize this difference.

For a point $\mathbf{x}' \in S'$, its predicted part label at the k-th level is determined by $\text{argmax}_m\{p_m^{(k)}(\mathbf{x}'_i), m = 1,\ldots,L^{(k)}\}$, where $p_m^{(k)}(\mathbf{x}'_i)$ is the m-th component of $\mathbf{p}^{(k)}(\mathbf{x}'_i)$. Using the predicted part labels of all points at the k-th level, we can partition S' into a set of parts, denoted $\{\mathcal{P}_1^{(k)},\cdots,\mathcal{P}_{L^{(k)}}^{(k)}\}$, where some sub-partitions can be empty. We call these parts a *pseudo-partition*. On S'', we also compute a pseudo-partition, denoted $\{\mathcal{Q}_1^{(k)},\cdots,\mathcal{Q}_{L^{(k)}}^{(k)}\}$.

For a pseudo-part $\mathcal{P}_l^{(k)}$, we define two statistical quantities: *belonging-confidence* and *outlier-confidence*, denoted by $\text{BC}(\mathcal{P}_l^{(k)})$ and $\text{OC}(\mathcal{P}_l^{(k)})$, respectively. The belonging-confidence measures the confidence with which points in $\mathcal{P}_l^{(k)}$ belong to $\mathcal{P}_l^{(k)}$ and the outlier-confidence measures the confidence with which the remaining points outside $\mathcal{P}_l^{(k)}$ do not belong to $\mathcal{P}_l^{(k)}$. They are defined as follows.

$$\begin{aligned}\text{BC}(\mathcal{P}_l^{(k)}, S') &= \text{MEAN}\{p_l^{(k)}(\mathbf{y}),\ \forall \mathbf{y} \in S' \cap \mathcal{P}_l^{(k)}\};\\ \text{OC}(\mathcal{P}_l^{(k)}, S') &= \text{MEAN}\{p_l^{(k)}(\mathbf{y}),\ \forall \mathbf{y} \in S' \backslash \mathcal{P}_l^{(k)}\}.\end{aligned} \tag{3}$$

As the pseudo-partitions of S' and S'' should be consistent with each other, we can impose the pseudo-partition of S' onto S'', *i.e.*, partition S'' according to the point assignment of $\{\mathcal{P}_1^{(k)},\cdots,\mathcal{P}_{L^{(k)}}^{(k)}\}$, and compute the corresponding belonging-confidence and the outlier-confidence values on S''. Because of self-consistency, we expect these values to be as close as possible to the corresponding values computed on S'. Similarly, we can also impose the pseudo-partition of S'' onto S' in a similar way. We call this type of consistency *part-level consistency*, and define the loss function as follows:

$$\begin{aligned}L_{\text{part}} = \sum_{k=1}^{K}\sum_{j=1}^{L^{(k)}}[\,&\alpha\|\,\text{BC}(\mathcal{P}_j^{(k)}, S') - \text{BC}(\mathcal{P}_j^{(k)}, S'')\|^2 + \\ &\beta\|\,\text{OC}(\mathcal{P}_j^{(k)}, S') - \text{OC}(\mathcal{P}_j^{(k)}, S'')\|^2 + \\ &\alpha\|\,\text{BC}(\mathcal{Q}_j^{(k)}, S'') - \text{BC}(\mathcal{Q}_j^{(k)}, S')\|^2 + \\ &\beta\|\,\text{OC}(\mathcal{Q}_j^{(k)}, S'') - \text{OC}(\mathcal{Q}_j^{(k)}, S')\|^2\,].\end{aligned} \tag{4}$$

Here α and β are dynamically adjusted: $\alpha = \beta = 1/2$ when the sub-partition appearing in the BC term is nonempty, otherwise we set $\alpha = 0$, $\beta = 1$. Fig. 2(b) illustrates part consistency on a chair model.

4.3 Hierarchical consistency

For a shape category possessing a part structure hierarchy, the semantic segmentation labels at different levels are strongly correlated. We propose *hierarchical consistency* to utilize this structure prior.

For a point $\mathbf{x} \in S$, we can use its probability vector at level $(k+1)$ to infer its part label probability at level k, *i.e.*, its parent level, just by merging the probability values of $\mathbf{p}^{(k+1)}(\mathbf{x})$ to form the probability vector at level k, according to the predefined shape hierarchy. For instance, in the chair structure, suppose the chair arm contains two parts: a vertical bar and a horizontal bar. We add the probability values of the vertical bar and horizontal bar together and set their sum as the probability value of the chair arm.

In this way, we can create a *pseudo-probability vector* for $\mathbf{x}$ at level k, denoted $\hat{\mathbf{p}}^{(k)}(\mathbf{x})$. Ideally $\hat{\mathbf{p}}^{(k)}(\mathbf{x}'_i)$ should be the same as $\mathbf{p}^{(k)}(\mathbf{x}''_i)$ predicted by the network and vice versa. We call this relation *hierarchical consistency*, and define a loss function on the points of S' and S''

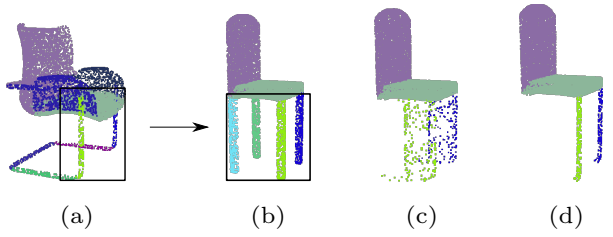

Fig. 3 Multilevel part substitution. The two front legs of the chair in (a) are selected to replace the four legs in (b). (c): unsatisfactory substitution by aligning the two regions directly. (d): good substitution by aligning the common parts (front legs) first. While the result is not physically plausible, it is suitable for training.

using KL-divergence as follows:

$$L_{\mathrm{h}} := \frac{1}{2n}\sum_{i=1}^{n}\sum_{k=1}^{K-1}\Big[D_{KL}\left(\hat{\mathbf{p}}^{(k)}(\mathbf{x}_i')\parallel \mathbf{p}^{(k)}(\mathbf{x}_i'')\right) + D_{KL}\left(\hat{\mathbf{p}}^{(k)}(\mathbf{x}_i'')\parallel \mathbf{p}^{(k)}(\mathbf{x}_i')\right)\Big]. \tag{5}$$

Fig. 2(c) illustrates hierarchical consistency on a chair model.

Note that the above hierarchical consistency is defined across two perturbed shapes. In fact, it is possible to impose hierarchical consistency on a single perturbed shape using $D_{KL}\big(\hat{\mathbf{p}}^{(k)}(\mathbf{x}_i')\parallel \mathbf{p}^{(k)}(\mathbf{x}_i')\big)$, but in practice we find that these consistency terms are easily satisfied as the multilevel probability vectors of the same shape are highly correlated, so do not give much assistance in semi-supervised training.

5 Multilevel part substitution

We propose a simple multilevel part-substitution algorithm to enrich the labeled 3D shapes for training. Given a randomly sampled labeled shape S, our algorithm executes the following steps to synthesize new shapes with geometry and structural variation.

Part selection is carried out first. We treat the hierarchical structure of shape S as a tree, where each shape part is a tree node. We visit each node from the coarsest level to the finest level. For a node at level k, a uniform random number in $[0, 1]$ is generated. If the number is smaller than a predefined threshold θ_k, we set the subtree under this node as a replacement candidate and stop visiting its children. Finally, we collect a set of subtrees to be replaced.

Next, part substitution is performed. For a part subtree P in the candidate list, we randomly select a subtree Q from those other shapes in which the root node of Q has the same semantic label as P's root node. Note that simple substitution of P by Q may result in strange-looking and partly-overlapping results (see Fig. 3(c)), so we replace P by Q as follows, to avoid unpleasant results as much as possible. We consider two cases.

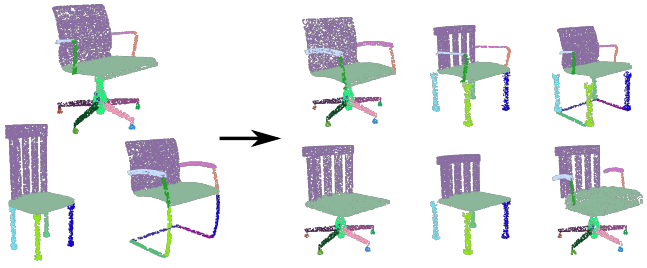

Fig. 4 Randomly generated shapes from three labeled chairs. Point colors correspond to semantic IDs at the finest level.

If the leaf nodes of P and Q have no common parts sharing the same semantics, we simply compute the affine transformation from the bounding box of Q to the bounding box of P and apply it to Q when replacing P.

However, if P and Q share some common semantic parts, denoted $P_s \in P$, $Q_s \in Q$, we align Q_s and P_s first to avoid odd results. The alignment transformation matrix is applied to Q directly. We also rescale the transformed Q to ensure that it is inside the original bounding box of S, to make the result visually plausible: see Fig. 3(d).

The θ_k values affect the degree of structure variation: frequent substitutions at the coarse level bring more structural variations. In our experiments, we set all θ_ks to 0.5. Fig. 4 shows a set of novel chairs synthesized from three chairs. More synthesized shapes used in our experiments are illustrated in Appendix C.

After generation, as all parts of the synthesized shapes inherit their original semantics, these shapes can be used as labeled data. Note that after part substitution, two different shape parts in a shape may overlap. We detect points inside these overlapping regions using a simple nearest neighbor search and do not use their labels during training, to avoid contradictions.

6 Network Design, Loss and Training Data

In this section, we present details of our network structure, loss function and training batches.

6.1 Network structure

We use an octree-based U-Net structure as our base network. The network is built upon the efficient open-source octree-based CNN [58, 59]. The U-Net structure has five and four levels of domain resolution, as illustrated in Fig. 5 and Appendix A, for object segmentation and scene segmentation, respectively.

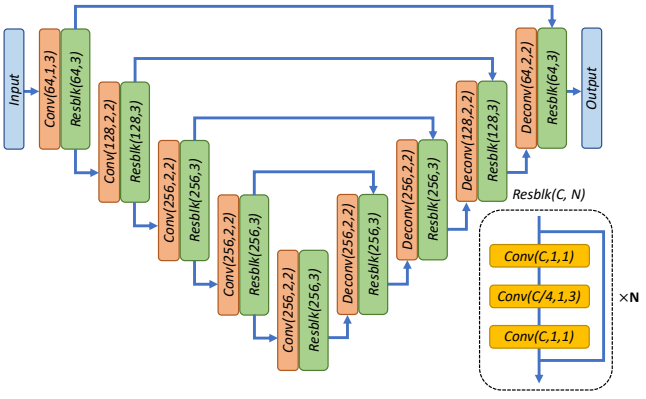

Fig. 5 Octree-based U-Net structure for shape segmentation. Conv(C, S, K) and Deconv(C, S, K) represent octree-based convolution and deconvolution. C, S, K are the number of output channels, stride, and kernel size. The network structure for indoor scene segmentation is provided in the appendix.

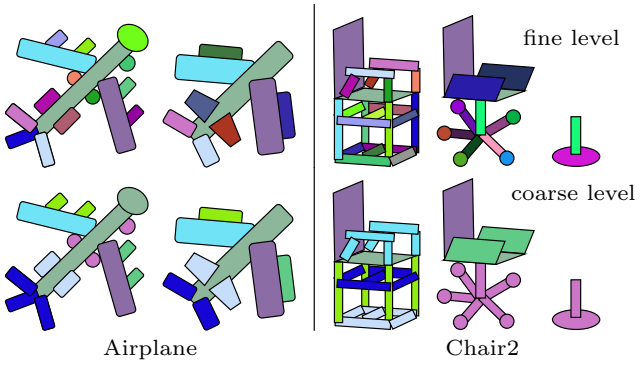

Fig. 6 The two-level fine-grained and hierarchical structures for our Airplane and Chair2 datasets. Unique colors at the fine level correspond to distinguishable shape parts, as a segmentation label. Several parts at the fine level are merged to form a unique segmentation label at the coarse level, and assigned the same color. Both Chair2 and Airplane have 8 different part labels at the coarse level, while at fine levels they have 36 and 20 part labels, respectively.

The maximum depths of the octree for 3D object segmentation and scene segmentation are 6 and 9, respectively. The input point cloud is converted to an octree first, whose nonempty finest octants store the average of the normals of the points within them. The point feature for a given point is found by trilinear interpolation within the octree. The numbers of network parameters for 3D object segmentation and 3D scene segmentation are 5.3×10^6 and 39.2×10^6, respectively. We call our network MCNet, for multi-consistency 3D deep learning network. In Section 7 we also demonstrate the efficiency of our approach based on other point-based backbones.

6.2 Loss function

Given a point cloud S in a training batch, the loss defined on its two randomly-perturbed copies S' and S'' is:

$$L_{\text{tc}} = \gamma L_{\text{seg}}(S', S'') + \lambda_{\text{pts}} L_{\text{point}} + \lambda_{\text{part}} L_{\text{part}} + \lambda_h L_{\text{h}}; \tag{6}$$

$\gamma = 0$ if S is an unlabeled point cloud.

6.3 Training batch construction

Half of the batch data is randomly selected from the labeled data, and the rest is randomly selected from the unlabeled dataset. If synthetic labeled data (Section 5) are available, half of the labeled data in the batch is selected from them, and the remainder is selected from the original labeled data. The labeled data in a batch may be duplicated if the labeled dataset is quite small. The network is trained from scratch with random initialization.

7 Experiments and Analysis

In this section, we demonstrate the efficacy and superiority of our semi-supervised approach on shape segmentation and scene segmentation, and an ablation study to validate our design.

Our experiments were conducted on a Linux server with a 3.6 GHz Intel Core I7-6850K CPU and a Tesla V100 GPU with 16 GB memory for experiments on shape objects, and a Tesla V100 with 32 GB memory for indoor scenes. We implemented our network using the TensorFlow framework [1].

7.1 Shape segmentation

7.1.1 Datasets

Our semi-supervised 3D segmentation approach was evaluated on the following datasets with different ratios of labeled data.

PartNet. The PartNet dataset [38] provides fine-grained, hierarchical segmentation of 26671 models in 24 object categories, and defines three levels (coarse, medium, fine) of segmentation for the benchmark.

Shape categories with customized hierarchy. We defined a two-level part hierarchy on two shape categories: the Chair from PartNet and the Airplane from ShapeNet [6], to further validate the effectiveness of our approach on other shape data and structural hierarchies. At the fine level, our new data provides finer-grained part labels than PartNet. For instance, each chair leg is treated as a different object part while all legs of a chair belong to a single part in the PartNet level-3 segmentation. The hierarchical relationship between each level is also differs from

Tab. 1 Segmentation results on PartNet. All metrics are averaged across 24 categories. **r** is the fraction of labeled data used for training.

		Coarse Level		Medium Level		Fine Level		Avg	
r	**Method**	**p-mIoU**	**s-mIoU**	**p-mIoU**	**s-mIoU**	**p-mIoU**	**s-mIoU**	**p-mIoU**	**s-mIoU**
2%	MIDNet	44.7	63.5	29.8	43.9	24.9	40.6	38.2	54.7
	MCNet	**47.6**	**68.4**	**33.4**	**48.9**	**27.4**	**45.8**	**41.3**	**60.0**
5%	MIDNet	53.2	69.1	35.9	48.4	32.7	46.0	46.7	60.7
	MCNet	**54.9**	**71.9**	**38.5**	**52.0**	**34.2**	**49.3**	**48.8**	**63.5**
10%	MIDNet	57.5	71.7	39.7	51.9	37.6	50.3	51.6	63.8
	MCNet	**60.9**	**75.5**	**43.9**	**54.6**	**40.2**	**52.0**	**54.8**	**67.0**
20%	MIDNet	64.2	75.7	44.6	55.4	43.3	54.2	57.7	68.0
	MCNet	**65.2**	**78.5**	**48.7**	**58.6**	**45.2**	**56.4**	**59.4**	**70.7**

PartNet. The hierarchical fine-grained structures of these two categories are illustrated in Fig. 6. To avoid confusion, we call our chair dataset Chair2. The Chair2 dataset contains 3303 models for training and 826 models for testing, and the Airplane dataset contains 1404 models for training and 366 models for testing.

ShapeNetPart. ShapeNetPart [71] contains 16 shape categories from ShapeNet. Each model is a point cloud with 2–6 part labels without a structural hierarchy.

ScanNet. The ScanNet dataset [11] contains 1613 3D indoor scenes with 20 labels for semantic segmentation. The numbers of scenes for training, validation, and testing are 1201, 312, and 100, respectively.

For the above datasets, we used a fixed seed to randomly pick a small fraction of the labeled training data, around 2%, and set it as the labeled data for semi-supervised training, and the remaining labeled training data was treated as unlabeled data in our training: no label information was utilized during training and part substitution. The original testing dataset was used as unseen test data for evaluating the trained network.

Each training batch contained 16 shapes. A maximum of 80000 iterations was used. We used the SGD optimizer with a learning rate of 0.1, decayed by a factor of 0.1 at the 40000-th and 60000-th iterations. For the loss function, we empirically set $\lambda_{\text{pts}} = \lambda_{\text{part}} = \lambda_h = 0.01$, via a simple grid search on the four biggest categories of PartNet. To conduct a statically meaningful evaluation, we ran training on each shape category three times with different randomly-selected labeled data, and report average results.

7.1.2 PartNet segmentation

On all 24 shape categories of PartNet, we experimented with our semi-supervised training scheme with different ratios of labeled data for our MCNet: 2%, 5%, 10%, and 20%. We generated as many randomly synthesized labeled shapes by part substitution as the number of original training data shapes. Following [38], we used the following metrics to evaluate the results.

p-mIoU. The IoU between the predicted point set and the ground-truth point set for each semantic part category is first computed over the test shapes, then the per-part-category IoUs are averaged. This metric helps evaluate how an algorithm performs for any given part category [38], but does not characterize the segmentation quality at the object level.

s-mIoU The part-wise IoU is first computed for each shape, then the mean IoU over all parts is computed on this shape, and finally, these mean IoUs are averaged over all test shapes. This metric is sensitive to missing ground-truth parts and the appearance of unwanted predicted parts in a shape.

We choose MIDNet [60] as a basis for comparison, which has unsupervised 3D pretraining with a fine tuning method and provides state-of-the-art results on PartNet with a small amount of labeled data. MIDNet was pretrained on the ShapeNet dataset. We fine tuned MIDNet with the same limited labeled data as our method, using the multilevel segmentation loss in Eq. (5). The results are reported in Table 1. Our MCNet achieved superior results to MIDNet on all tests at all segmentation levels.

In Fig. 7, we illustrate segmentation results and error maps resulting from our approach and MIDNet on a set of test shapes, trained with 2% labeled data. The results clearly show that our method has lower segmentation error.

We also replaced our octree-based CNN backbone with other popular point-based deep learning frameworks: PointNet++ [44] and PointCNN [35],

Tab. 2 Segmentation results on PartNet with different backbone networks. All metrics are averaged across 3 levels of the test dataset for 24 categories. **r** is the proportion of labeled data used for training. Baseline is the supervised approach with multilevel segmentation loss. Ours is the backbone with our semi-supervised approach.

		PointNet++		PointCNN		OCNN	
r	**Method**	**p-mIoU**	**s-mIoU**	**p-mIoU**	**s-mIoU**	**p-mIoU**	**s-mIoU**
2%	Baseline	34.6	47.6	35.3	50.6	35.8	51.2
	Ours	39.7	57.7	40.7	58.3	41.3	60.0
5%	Baseline	42.3	53.6	42.9	55.2	42.8	54.7
	Ours	47.8	62.6	49.2	62.2	48.8	63.5
10%	Baseline	48.0	59.2	49.1	61.0	49.4	60.9
	Ours	51.5	65.1	52.3	65.7	54.8	67.0
20%	Baseline	53.5	63.9	54.0	63.8	56.2	66.4
	Ours	56.2	68.6	56.5	68.3	59.4	70.7

Tab. 3 Segmentation results on the test dataset containing Chair2 and Airplane. **r** is the proportion of labeled data used for training.

		Fine Level		Coarse Level	
r	**Method**	**p-mIoU**	**s-mIoU**	**p-mIoU**	**s-mIoU**
2%	MIDNet	75.7	85.6	87.9	87.4
	MCNet	**82.4**	**89.0**	**91.4**	**91.4**
5%	MIDNet	81.0	87.5	90.1	89.0
	MCNet	**84.5**	**90.2**	**92.1**	**92.1**
10%	MIDNet	82.8	88.0	90.4	89.9
	MCNet	**85.7**	**90.5**	**92.3**	**92.2**
20%	MIDNet	83.5	89.4	90.9	90.8
	MCNet	**86.0**	**91.1**	**92.2**	**92.7**

and tested their segmentation performance. Table 2 reports the significant improvements brought by our multilevel consistency and part substitution, compared to their purely-supervised baseline. We also found that these backbones did not yield better results than the octree-based CNN backbone.

7.1.3 Segmentation on shape categories with customized hierarchy

Like the experiments on PartNet, the experiments on Chair2 and Airplane also showed that our approach is significantly better than MIDNet (see Table 3). Several segmentation results are also illustrated in Fig. 7.

7.1.4 ShapeNetPart segmentation

As there is no structural hierarchy in ShapeNetPart, hierarchy consistency loss was dropped from our loss function. We report the mean IoU across all categories (c-mIoU) and across all instances (i-mIoU), commonly used metrics in the ShapeNetPart segmentation benchmark. Table 4 compares results from various methods using 5% labeled data for training. It is clear that our method is superior to others, while MIDNet is second best. We also made a more thorough comparison to MIDNet using other ratios of labeled training data. The results in Table 5 show that MCNet always performed much better than MIDNet.

Tab. 4 Segmentation results for different methods on ShapeNetPart, with 5% labeled training data.

Method	**c-mIoU**	**i-mIoU**
SO-Net [33]	-	69.0
PointCapsNet [74]	-	70.0
MortonNet [52]	-	77.1
JointSSL [3]	-	77.4
Multi-task [24]	72.1	77.7
ACD [18]	-	79.7
MIDNet [60]	77.7	80.7
MCNet	**79.8**	**82.2**

We also conducted a few-shot experiment by

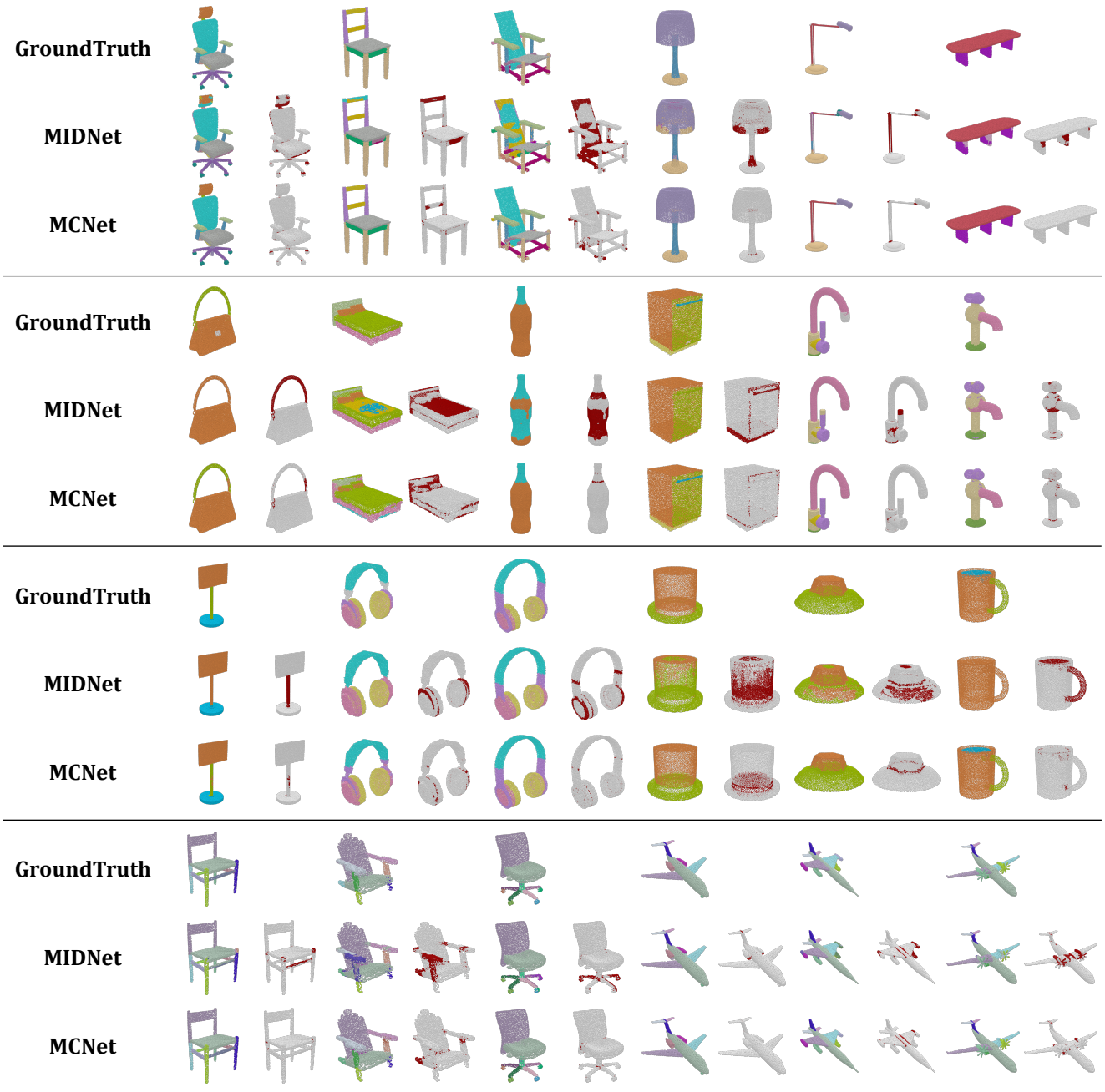

Fig. 7 Fine-level segmentation results from our MCNet and MIDNet. 2% labeled data were used in training. Point colors in the segmentation results correspond to part ID. In the error maps alongside the segmentation results, red points indicate wrongly predicted labels. The top three sets of examples came from the PartNet test set; those at the bottom came from the Chair2 and Airplane test data.

following the setting in the state-of-the-art few-shot 3D segmentation method [57]: eight categories of ShapeNetPart were tested and 10 labeled shapes used for training. We trained our network 5 times and sampled 10 labeled shapes randomly from the original dataset each time, reporting average results in Table 6. It shows that our method is superior to that of [57] for most of the tested categories.

7.2 ScanNet segmentation

7.2.1 Setting

We have also applied our semi-supervised method to indoor semantic scene segmentation. We choose the ScanNet dataset [11] as a testbed, and used 1%, 5%, 10%, and 20% labeled scenes from the original training dataset, with the remainder of the training dataset regarded as unlabeled data. For a fair comparison,

Tab. 5 Segmentation results for ShapeNetPart. Higher mIoU values are better. **r** is the proportion of labeled data used for training.

r	Method	c-mIoU	i-mIoU
2%	MIDNet	73.9	78.4
	MCNet	**76.1**	**81.2**
5%	MIDNet	77.7	80.7
	MCNet	**79.8**	**82.2**
10%	MIDNet	79.2	82.3
	MCNet	**81.8**	**84.2**
20%	MIDNet	81.7	83.1
	MCNet	**83.0**	**84.3**

Tab. 6 mIoU results for our method and that of [57], for eight shape categories selected by [57].

Category	[57]	Ours
Airplane	67.3	**73.9**
Bag	74.4	**81.7**
Cap	**86.3**	84.4
Chair	83.4	**87.2**
Lamp	68.7	**76.5**
Laptop	93.8	**95.4**
Mug	90.9	**95.5**
Table	74.2	**74.8**
Mean	79.8	**83.7**

we used the labeled scenes from [25]. We measured mean IoU to evaluate segmentation quality on the validation set. As the ScanNet dataset does not provide hierarchical segmentation, we defined a 2-level segmentation on ScanNet: classes at the fine level are the original segmentation classes; at the coarse level, we merged the semantic classes into 6 categories using semantic affinity according to WordNet [14], calling this hierarchy HW. Details of these hierarchies are presented in Appendix B. As our part substitution is not intended for 3D scenes, we did not synthesize 3D labeled scenes for training.

7.2.2 Data perturbation

We used the same augmentation configuration as [25]: a random rotation with pitch, yaw, and roll angles less than $3^\circ, 180^\circ, 3^\circ$, respectively, a uniform scaling in the range $[0.9, 1.1]$, random translations along x-, y- axes within the range $[0.8, 1.2]$, and a color transformation including auto contrast, color translation and color jitter. We randomly sampled 20% points from a scene in each training iteration.

Tab. 7 Fine level segmentation results for ScanNet. **r** is the proportion of labeled data used for training. −H means without hierarchy, +H means with hierarchy. The model from [25] was used to generate its segmentation results.

r	Method	mIoU
1%	[25]	29.3
	Our supervised baseline	27.0
	MCNet−H ($\lambda_{\text{pts}} = \lambda_{\text{part}} = 0.005$)	28.7
	MCNet+H ($\lambda_{\text{pts}} = \lambda_{\text{part}} = \lambda_h = 0.005$)	**29.4**
5%	[25]	45.4
	Our supervised baseline	47.9
	MCNet−H ($\lambda_{\text{pts}} = \lambda_{\text{part}} = 0.05$)	48.2
	MCNet+H ($\lambda_{\text{pts}} = \lambda_{\text{part}} = \lambda_h = 0.05$)	**48.3**
10%	[25]	59.5
	Our supervised baseline	58.1
	MCNet−H ($\lambda_{\text{pts}} = \lambda_{\text{part}} = 0.1$)	59.1
	MCNet+H ($\lambda_{\text{pts}} = \lambda_{\text{part}} = \lambda_h = 0.1$)	**60.3**
20%	[25]	64.1
	Our supervised baseline	62.8
	MCNet−H ($\lambda_{\text{pts}} = \lambda_{\text{part}} = 0.1$)	63.9
	MCNet+H ($\lambda_{\text{pts}} = \lambda_{\text{part}} = \lambda_h = 0.1$)	**64.9**

Tab. 8 Fine level segmentation results for ScanNet using a different customized hierarchy. **r** is the proportion of labeled data used for training.

r	HW	HA	HB	HC
1%	29.4	29.2	29.7	29.5
5%	48.3	48.9	48.6	48.4
10%	60.3	60.2	60.2	60.0
20%	64.9	64.8	64.4	64.7

7.2.3 Parameters and training protocol

Each batch contained 4 shapes, with two from the labeled scenes and two from the unlabeled scenes. A maximum of 60000 iterations was used. We used the SGD optimizer with a learning rate of 0.1, decayed at the 30000-th and 45000-th iterations by a factor of 0.1. We tried different multilevel consistency weights, and found that smaller weights improve results when the proportion of labeled data is low. The optimal settings we found are reported alongside the network results in Table 7.

7.2.4 Results

We compared our supervised baseline, *i.e.*, using the segmentation loss and labeled data only, our method with and without structural hierarchy loss, and the state-of-the-art unsupervised pretraining with fine-tuning method proposed by [25]. As Table 7 shows, our

Tab. 9 Numbers of shapes in the four shape categories from PartNet used in our ablation study.

	Chair	Lamp	Storage	Table
Labeled	90	31	32	114
Unlabeled	4399	1523	1556	5593
Test	1217	419	451	1668

supervised baseline and our method without using the hierarchy loss worked less well than [25] except in the test with 5% labeled data. With the additional hierarchy loss, our method performed best in all tests.

7.2.5 Sensitivity to customized hierarchy

To study whether our method on ScanNet is sensitive to the customized hierarchy, we randomly grouped fine level parts into 6 categories three times, and created three different 2-level hierarchies, HA, HB, HC. Table 8 reports the segmentation results using these customized hierarchies. We find that MCNet achieves similar results using HW, HA, HB, and HC, so conclude that while our approach benefits from hierarchical relationships, it is insensitive to the hierarchy construction.

7.3 Ablation study

We next evaluate the efficacy of our consistency loss, part substitution, and hyper-parameter selection approaches, using the four biggest categories from PartNet: Chair, Table, Storage, and Lamp, as our testbed. We used 2% labeled data here only. Results are reported in Table 9. The network was trained for each shape category individually.

7.3.1 Multilevel consistency loss and part substitution

We designed a series of ablation studies to validate the advantage of our multilevel consistency losses and part substitution, using as baseline our network trained with the multilevel segmentation loss on the limited labeled data only. Table 10 reports results for the baseline (ID-(1)) and the baseline with different combinations of our multilevel consistency losses with semi-supervised training. We state how many labeled shapes were synthesized via multilevel part substitution from the 2% labeled data used for training; N indicates the total number of labeled and unlabeled data items.

Experiments (2)–(4) clearly show that utilization of any consistency loss can improve segmentation accuracy. Experiment (5) indicates that synthesizing labeled shapes by part substitution can significantly improve the network results even when used in a purely supervised training manner. Combinations of different types of consistency losses (6)–(9) further boost network performance; combining all consistency losses in (9) works best. Adding synthesized labeled shapes (10)–(12) further helps the network to reach its highest accuracy. The configuration in (12) is the default and optimal setting of MCNet used in Section 7.1.

Concurrent work to this work, PointCutMix [72] proposes a data augmentation method which finds the optimal assignment between two labeled point clouds and generates new training data by replacing points in one sample with their optimally assigned pairs. We implemented their approach and used the generated shapes to enhance training. A few synthesized shapes are illustrated in Appendix C. Experiments (13) and (14) show that their data augmentation method can enhance the network accuracy to a certain degree, but does not bring as significant an improvement as our approach, due to its lack of awareness of part structures during data synthesis.

7.3.2 Data perturbation

We also examined how data perturbation affects the performance of MCNet. The experimental setup was as in Section 7.3.1; we only varied the ranges of rotation, scaling, and translation data perturbation parameters, with results shown in Table 11. Configuration (1) is our default configuration.

By varying the range of the random rotation angle, we found that when using a smaller or larger angle range, the network performance slightly decreases: see (1)–(3).

Tests (1), (4), and (5) reveal that an appropriate shape scaling is important. Note that any part outside the unit sphere caused by a large scaling is removed by our perturbation, so there are fewer corresponding points between the two perturbed shape copies used in our consistency loss computation.

Tests (1), (6), and (7) also show that an appropriate translation helps our training. Making translation too large also results in missing shape geometry, degrading the efficacy of our consistency loss.

8 Conclusions

We have presented an effective semi-supervised approach for 3D shape segmentation. Our novel multilevel consistency and part substitution scheme harnesses the structural consistency hidden in both unlabeled data and labeled data, for our network training, leading to superior performance on 3D segmentation tasks with few labeled data items. We believe that our multilevel consistency will find more applications, potentially being useful for semi-supervised

Tab. 10 Ablation study for MCNet trained on four categories from PartNet using different loss combinations and synthesized shapes, and 2% labeled data. ✓ indicates that the corresponding loss was employed during training. Aug is the number of synthesized shapes used. N is the total number of labeled and unlabeled data items. In (13), (14), we used the method of [72] to generate augmented labeled shapes for training. Quality metrics were measured on the test dataset.

Experimental configuration						Coarse Level		Medium Level		Fine Level		Avg	
ID	L_{seg}	L_{point}	L_{part}	L_{h}	Aug	p-mIoU	s-mIoU	p-mIoU	s-mIoU	p-mIoU	s-mIoU	p-mIoU	s-mIoU
(1)	✓				0	37.7	56.2	25.6	34.0	21.8	30.3	28.3	40.2
(2)	✓	✓			0	40.3	59.8	27.6	37.1	23.4	33.6	30.4	43.5
(3)	✓		✓		0	40.5	57.9	26.4	35.3	22.5	31.1	29.8	41.4
(4)	✓			✓	0	39.9	59.1	27.1	37.7	23.3	33.6	30.1	43.5
(5)	✓				N	41.2	62.1	28.7	40.4	24.2	36.2	31.3	46.2
(6)	✓	✓	✓		0	41.3	61.3	27.7	39.8	23.6	35.9	30.9	45.6
(7)	✓	✓		✓	0	40.5	62.4	27.7	40.9	23.5	36.9	30.6	46.7
(8)	✓		✓	✓	0	41.3	60.6	27.6	38.6	23.5	34.6	30.8	44.6
(9)	✓	✓	✓	✓	0	42.7	62.7	28.4	41.5	24.2	37.5	31.7	47.2
(10)	✓	✓	✓	✓	$N/4$	42.8	64.7	29.7	43.7	26.0	39.5	32.8	49.3
(11)	✓	✓	✓	✓	$N/2$	42.8	65.3	30.3	44.1	26.1	**39.9**	33.1	49.8
(12)	✓	✓	✓	✓	N	**43.1**	**65.6**	**30.5**	**44.2**	**26.3**	**39.9**	**33.3**	**49.9**
(13)	✓				N [72]	38.2	54.6	27.7	35.0	23.4	31.6	29.8	40.4
(14)	✓	✓	✓	✓	N [72]	40.4	59.8	30.1	39.9	24.9	36.2	31.8	45.3

Tab. 11 Ablation study for MCNet trained on four categories from PartNet under different data perturbation configurations, with 2% labeled data. Quality metrics were measured on the test dataset.

Perturbation configuration				Coarse Level		Medium Level		Fine Level		Avg	
ID	rotation	scaling	translation	p-mIoU	s-mIoU	p-mIoU	s-mIoU	p-mIoU	s-mIoU	p-mIoU	s-mIoU
(1)	[-10°, 10°]	[0.75, 1.25]	[-0.25, 0.25]	43.1	**65.6**	**30.5**	**44.2**	**26.3**	**39.9**	**33.3**	**49.9**
(2)	[-5°, 5°]	[0.75, 1.25]	[-0.25, 0.25]	43.0	65.4	30.4	43.8	25.9	39.6	33.1	49.6
(3)	[-20°, 20°]	[0.75, 1.25]	[-0.25, 0.25]	41.6	**65.6**	**30.5**	43.6	26.1	39.8	32.7	49.7
(4)	[-10°, 10°]	[0.90, 1.10]	[-0.25, 0.25]	42.7	64.5	29.9	43.1	25.7	39.1	32.7	48.9
(5)	[-10°, 10°]	[0.60, 1.40]	[-0.25, 0.25]	**43.3**	64.8	30.3	43.4	26.0	39.5	33.2	49.2
(6)	[-10°, 10°]	[0.75, 1.25]	[-0.10, 0.10]	41.5	64.8	29.7	43.4	25.5	39.3	32.2	49.2
(7)	[-10°, 10°]	[0.75, 1.25]	[-0.40, 0.40]	41.7	64.6	29.8	43.1	25.6	39.2	32.3	49.0

image segmentation.

There are still a few unexplored directions. Firstly, it is possible to extend the hierarchical consistency from points to parts and involve more structural levels (> 3) to improve the training, which may be especially beneficial for more complicated datasets and part structures. Secondly, it would be helpful to synthesize novel shapes and scenes from both labeled and unlabeled data with more diverse structural and geometry variations for semi-supervised learning.

9 Declarations

9.1 Availability of data and materials

PartNet, ShapeNetPart, and ScanNet are all publicly released datasets. Our shape categories with the customized part hierarchy are also available.

9.2 Competing interests

The authors declare that they have no competing interests.

9.3 Authors' contributions

Chunyu Sun proposed and implemented the key idea, conducted the main experiments, and contributed to paper writing. Yuqi Yang contributed to the comparison of unsupervised pretraining. Xin Tong and Haoxiang

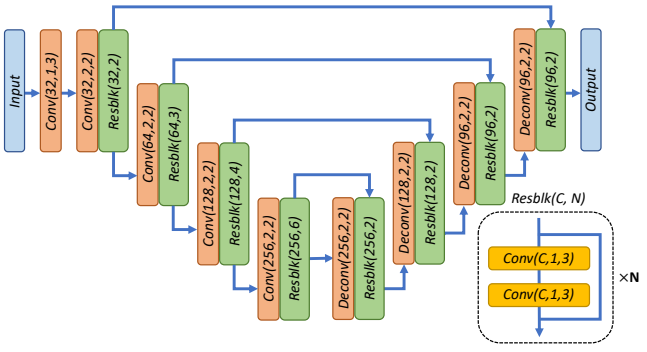

Fig. 8 Octree-based U-Net structure for scene segmentation. Conv(C, S, K) and Deconv(C, S, K) represent octree-based convolution and deconvolution. C, S, K are the number of output channels, stride, and kernel size.

Guo contributed to the part substitution algorithm. Pengshuai Wang provided guidance on the design of the O-CNN backbone and data augmentation. Xin Tong and Heung-Yeung Shum supervised the findings of this work and verified the key idea. Yang Liu led the project and contributed to the key idea, experimental design, and paper writing.

Appendices

A. Network structure for scene segmentation

An octree-based U-Net structure is used as our base network. It has four levels of domain resolution: see Fig. 8. The maximum octree depth is 9.

B. ScanNet hierarchy

The coarse levels of HW, HA, HB, and HC are shown in Table 12. Fine classes are merged to the coarse level. Numbers in the table give the coarse label ID.

Tab. 12 Customized hierarchies for ScanNet. The number is the coarse label ID.

Category	HW	HA	HB	HC
Wall	1	6	4	4
Floor	2	3	6	3
Cabinet	3	5	4	2
Bed	3	3	2	5
Chair	3	2	3	3
Sofa	3	4	5	4
Table	3	3	3	5
Door	4	4	5	1
Window	1	2	6	4
Bookshelf	4	2	5	5
Picture	4	1	6	2
Counter	3	2	6	1
Desk	3	2	5	1
Curtain	5	1	2	2
Refrigerator	4	1	4	2
Shower curtain	5	2	1	2
toilet	6	4	4	1
sink	6	6	6	2
bathtub	5	5	6	5
Other furniture	4	1	3	2

C. Data augmentation by part substitution

In Figs. 9 to 11, we illustrate a sample set of shapes augmented by part substitution on 2% labeled data. The majority of the augmented shapes are plausible and would help to enrich the labeled data for network training. In Fig. 12, we render the augmented shapes generated by the approach of [72]. We can observe many implausible shape parts which do not assist training.

References

[1] M. Abadi, A. Agarwal, P. Barham, and et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. 7

[2] I. Alhashim, H. Li, K. Xu, J. Cao, R. Ma, and H. Zhang. Topology-varying 3D shape creation via structural blending. *ACM Trans. Graph.*, 33(4), 2014. 3

[3] A. Alliegro, D. Boscaini, and T. Tommasi. Joint supervised and self-supervised learning for 3D real world challenges. In *International Conference on Pattern Recognition (ICPR)*, pages 6718–6725, 2021. 9

[4] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013. 2

[5] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel. MixMatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019. 3

[6] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An information-rich 3D model repository. arXiv:1512.03012 [cs.GR], 2015. 2, 7

[7] S. Chaudhuri, E. Kalogerakis, L. Guibas, and V. Koltun. Probabilistic reasoning for assembly-based 3D modeling. *ACM Trans. Graph.*, 30(4), 2011. 3

[8] Y. Chen, V. T. Hu, E. Gavves, T. Mensink, P. Mettes, P. Yang, and C. G. Snoek. PointMixup: Augmentation for point clouds. In *ECCV*, pages 330–345, 2020. 3

[9] Z. Chen, K. Yin, M. Fisher, S. Chaudhuri, and H. Zhang. BAE-NET: Branched autoencoder for shape co-segmentation. In *ICCV*, 2019. 2, 3

[10] C. Choy, J. Gwak, and S. Savarese. 4D spatio-temporal ConvNets: Minkowski convolutional neural networks. In *CVPR*, 2019. 2

[11] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017. 8, 10

[12] A. Dai and M. Nießner. 3dmv: Joint 3d-multi-view

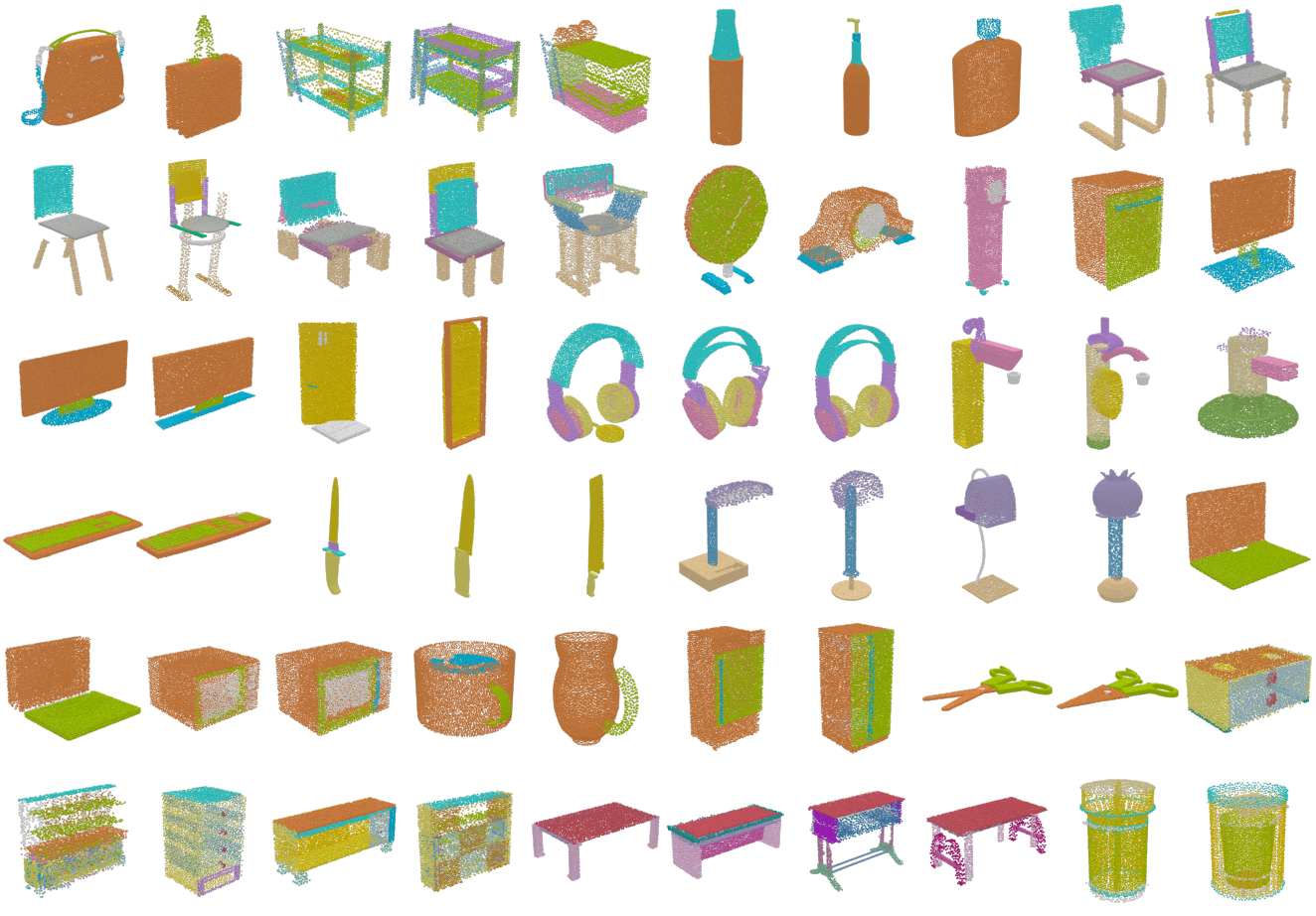

Fig. 9 Augmented shapes based on 2% labeled data from PartNet.

prediction for 3d semantic scene segmentation. In *ECCV*, pages 452–468, 2018. 2

[13] B. Deng, K. Genova, S. Yazdani, S. Bouaziz, G. Hinton, and A. Tagliasacchi. CvxNets: Learnable convex decomposition. In *CVPR*, 2020. 2

[14] C. Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998. 11

[15] G. French, S. Laine, T. Aila, M. Mackiewicz, and G. Finlayson. Semi-supervised semantic segmentation needs strong, varied perturbations. In *BMVC*, 2020. 3

[16] Q. Fu, X. Chen, X. Su, and H. Fu. Pose-inspired shape synthesis and functional hybrid. *IEEE Trans. Vis. Comput. Graphics*, 23(12):2574–2585, 2017. 3

[17] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin. Modeling by example. *ACM Trans. Graph.*, 23(3):652–663, 2004. 3

[18] M. Gadelha, A. RoyChowdhury, G. Sharma, E. Kalogerakis, L. Cao, E. Learned-Miller, R. Wang, and S. Maji. Label-efficient learning on point clouds using approximate convex decompositions. In *ECCV*, pages 473–491. Springer, 2020. 9

[19] K. Genova, F. Cole, A. Sud, A. Sarna, and T. Funkhouser. Deep structured implicit functions. In *CVPR*, 2020. 2

[20] B. Graham, M. Engelcke, and L. van der Maaten. 3D semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, 2018. 2

[21] Y. Guan, H. Liu, K. Liu, K. Yin, R. Hu, O. van Kaick, Y. Zhang, E. Yumer, N. Carr, R. Mech, , and H. Zhang. FAME: 3D shape generation via functionality-aware model evolution. *IEEE Trans. Vis. Comput. Graphics*, 2020. 3

[22] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun. Deep learning for 3D point clouds: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. 2

[23] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or. MeshCNN: A Network with an Edge. *ACM Trans. Graph.*, 38(4):90:1–90:12, 2019. 2

[24] K. Hassani and M. Haley. Unsupervised multi-task feature learning on point clouds. In *ICCV*, 2019. 2, 9

[25] J. Hou, B. Graham, M. Nießner, and S. Xie. Exploring data efficient 3D scene understanding with contrastive scene contexts. In *CVPR*, 2021. 2, 11, 12

[26] H. Huang, E. Kalogerakis, and B. Marlin. Analysis and synthesis of 3D shape families via deep-learned generative models of surfaces. *Comput. Graph. Forum*, 34:25–38, 2015. 3

[27] S.-S. Huang, Z.-Y. Ma, T.-J. Mu, H. Fu, and S.-M.

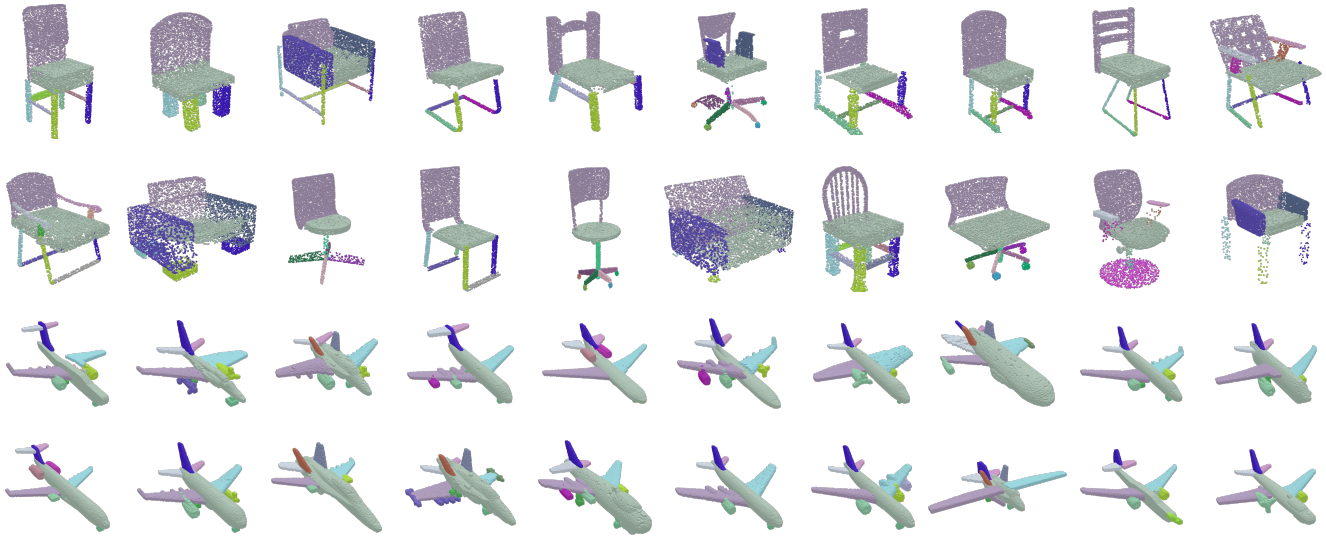

Fig. 10 Augmented shapes based on 2% labeled data from Chair2 and Airplane.

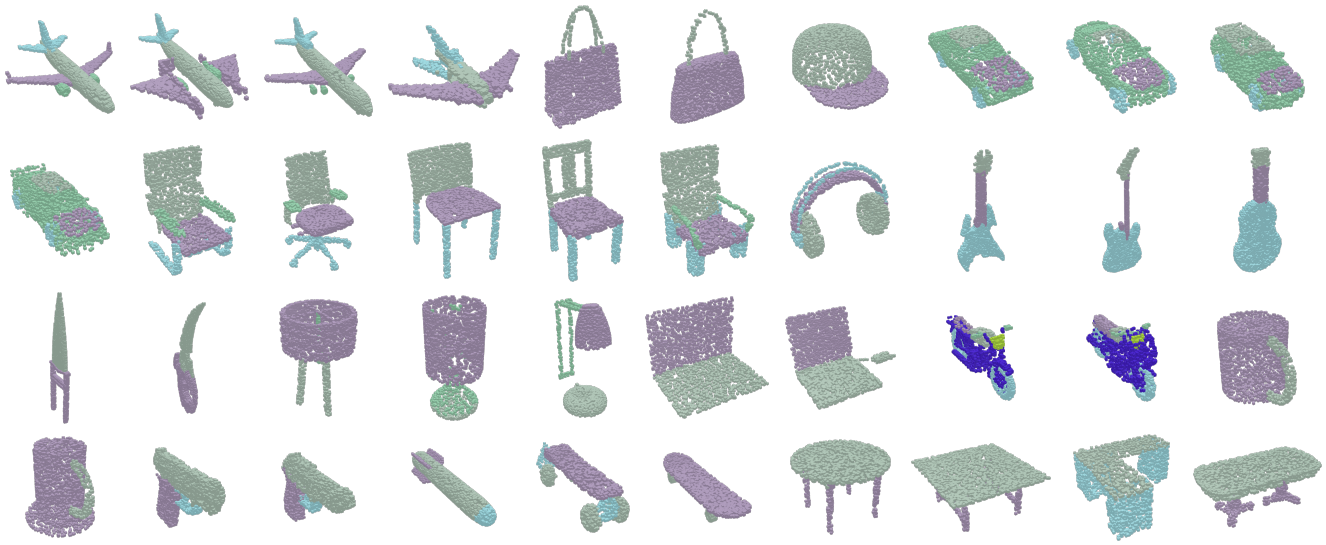

Fig. 11 Illustration of the augmented shapes based on 2% labeled data of ShapeNetPart.

Hu. Supervoxel Convolution for Online 3D Semantic Segmentation. *ACM Trans. Graph.*, 40(3), 2021. 2

[28] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri. 3D shape segmentation with projective convolutional networks. In *CVPR*, 2017. 2

[29] E. Kalogerakis, A. Hertzmann, and K. Singh. Learning 3D mesh segmentation and labeling. *ACM Trans. Graph.*, 29(4):102:1–102:12, 2010. 2

[30] Z. Ke, D. Qiu, K. Li, Q. Yan, and R. W. Lau. Guided collaborative training for pixel-wise semi-supervised learning. In *ECCV*, 2020. 1, 3

[31] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017. 3, 5

[32] D. Lee, J. Lee, J. Lee, H. Lee, M. Lee, S. Woo, and S. Lee. Regularization strategy for point cloud via rigidly mixed sample. In *CVPR*, pages 15900–15909, 2021. 3

[33] J. Li, B. M. Chen, and G. H. Lee. SO-Net: Self-organizing network for point cloud analysis. In *CVPR*, pages 9397–9406, 2018. 9

[34] R. Li, X. Li, P.-A. Heng, and C.-W. Fu. PointAugment: an auto-augmentation framework for point cloud classification. In *CVPR*, pages 6378–6387, 2020. 3

[35] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. PointCNN: Convolution on X-transformed points. In *NeurIPS*, pages 820–830, 2018. 2, 8

[36] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst. Geodesic Convolutional Neural Networks on Riemannian Manifolds. In *ICCV*, 2015. 2

[37] K. Mo, P. Guerrero, L. Yi, H. Su, P. Wonka, N. Mitra, and L. J. Guibas. StructEdit: Learning structural shape variations. In *CVPR*, 2020. 3

[38] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su. PartNet: A large-scale benchmark

Fig. 12 Augmented shapes based on 2% labeled data from PartNet using PointCutMix [72].

for fine-grained and hierarchical part-level 3D object understanding. In *CVPR*, 2019. 7, 8

[39] S. Muralikrishnan, V. G. Kim, and S. Chaudhuri. Tags2Parts: Discovering semantic regions from shape tags . In *CVPR*, 2018. 2

[40] Y. Ouali, C. Hudelot, and M. Tami. Semi-Supervised Semantic segmentation with cross-consistency training. In *CVPR*, 2020. 1, 3

[41] D. Paschalidou, A. O. Ulusoy, and A. Geiger. Superquadrics revisited: Learning 3D shape parsing beyond cuboids. In *CVPR*, 2019. 2

[42] A. Poulenard and M. Ovsjanikov. Multi-directional geodesic neural networks via equivariant convolution. *ACM Trans. Graph.*, 37(6):236:1–236:14, 2018. 2

[43] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017. 2

[44] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 2, 8

[45] R. S. V. Rodrigues, J. F. M. Morgado, and A. J. P. Gomes. Part-based mesh segmentation: A survey. *Comput. Graph. Forum*, 37(6):235–274, 2018. 2

[46] A. Shamir. A survey on mesh segmentation techniques. *Comput. Graph. Forum*, 27(6):1539–1556, 2008. 2

[47] G. Sharma, E. Kalogerakis, and S. Maji. Learning point embeddings from shape repositories for few-shot segmentation. In *3DV*, 2019. 2

[48] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. C. E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel. FixMatch: Simplifying Semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020. 3

[49] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, pages 1746–1754, 2017. 2

[50] C.-Y. Sun, Q.-F. Zou, X. Tong, and Y. Liu. Learning Adaptive hierarchical cuboid abstractions of 3D shape collections. *ACM Trans. Graph.*, 38(6):241:1–241:13, 2019. 2

[51] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017. 3, 5

[52] A. Thabet, H. Alwassel, and B. Ghanem. Self-Supervised Learning of Local Features in 3D Point Clouds. In *CVPR Workshops*, 2020. 9

[53] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas. KPConv: Flexible and deformable convolution for point clouds. In *ICCV*, 2019. 2

[54] S. Tulsiani, , H. Su, L. J. Guibas, A. A. Efros, and J. Malik. Learning shape abstractions by assembling volumetric primitives. In *CVPR*, 2017. 2

[55] J. E. van Engelen and H. H. Hoos. A survey on semi-supervised learning. *Machine Learning*, 109:373–440, 2020. 3

[56] K. Wang, B. Zhan, C. Zu, X. Wu, J. Zhou, L. Zhou, and Y. Wang. Tripled-uncertainty guided mean teacher model for semi-supervised medical image segmentation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, pages 450–460, 2021. 3

[57] L. Wang, X. Li, and Y. Fang. Few-shot learning of part-specific probability space for 3D shape segmentation. In *CVPR*, 2020. 3, 10, 11

[58] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong. O-CNN: Octree-based convolutional neural networks for 3D shape analysis. *ACM Trans. Graph.*, 36(4):72:1–72:11, 2017. 2, 6

[59] P.-S. Wang, Y. Liu, and X. Tong. Deep Octree-based CNNs with output-guided skip connections for 3D shape and scene completion. In *CVPR workshop*, 2020. 6

[60] P.-S. Wang, Y.-Q. Yang, Q.-F. Zou, Z. Wu, Y. Liu, and X. Tong. Unsupervised 3D learning for shape analysis via multiresolution instance discrimination. In *AAAI*, 2020. 2, 4, 8, 9

[61] X. Wang, B. Zhou, H. Fang, X. Chen, Q. Zhao, and K. Xu. Learning to group and label fine-grained shape components. *ACM Trans. Graph.*, 37(6):210:1–210:14, 2018. 2

[62] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.*, 38(5):146:1–146:12, 2019. 2

[63] R. Wu, Y. Zhuang, K. Xu, H. Zhang, and B. Chen. PQ-NET: A generative part Seq2Seq network for 3D shapes. In *CVPR*, 2020. 3

[64] S. Xie, J. Gu, D. Guo, C. R. Qi, L. J. Guibas, and

O. Litany. PointContrast: Unsupervised pre-training for 3D point cloud understanding. *ECCV*, 2020. 2

[65] X. Xie, K. Xu, N. J. Mitra, D. Cohen-Or, W. Gong, Q. Su, and B. Chen. Sketch-to-Design: Context-Based Part Assembly. *Comput. Graph. Forum*, 32:233–245, 2013. 3

[66] Z. Xie, K. Xu, W. Shan, L. Liu, Y. Xiong, and H. Huang. Projective feature learning for 3D shapes with multi-view depth images. *Comput. Graph. Forum*, 34(7):1–11, 2015. 2

[67] K. Xu, V. G. Kim, Q. Huang, and E. Kalogerakis. Data-driven shape analysis and processing. *Comput. Graph. Forum*, 36(1):101–132, 2015. 2

[68] K. Xu, H. Zhang, D. Cohen-Or, and B. Chen. Fit and diverse: Set evolution for inspiring 3D shape galleries. *ACM Trans. Graph.*, 31(4), 2012. 3

[69] X. Xu and G. H. Lee. Weakly supervised semantic point cloud segmentation: Towards 10 × fewerlabels. In *CVPR*, 2020. 2

[70] Y. Yang, S. Liu, H. Pan, Y. Liu, and X. Tong. FCNN: Convolutional Neural networks on 3D surfaces using parallel frames. In *CVPR*, 2020. 2

[71] L. Yi, L. Guibas, A. Hertzmann, V. G. Kim, H. Su, and E. Yumer. Learning hierarchical shape segmentation and labeling from online repositories. *ACM Trans. Graph.*, 36(4):70:1–70:12, 2017. 2, 8

[72] J. Zhang, L. Chen, B. Ouyang, B. Liu, J. Zhu, Y. Chen, Y. Meng, and D. Wu. PointCutMix: Regularization Strategy for Point Cloud Classification. arXiv:2101.01461, 2021. 3, 12, 13, 14, 17

[73] J. Zhang, C. Zhu, L. Zheng, and K. Xu. Fusion-aware point convolution for online semantic 3d scene segmentation. In *CVPR*, 2020. 2

[74] Y. Zhao, T. Birdal, H. Deng, and F. Tombari. 3D Point Capsule Networks. In *CVPR*, 2019. 9

[75] Y. Zheng, D. Cohen-Or, and N. J. Mitra. Smart variations: Functional substructures for part compatibility. *Comput. Graph. Forum*, 32:195–204, 2013. 3

[76] C. Zhu, K. Xu, S. Chaudhuri, L. Yi, L. J. Guibas, and H. Zhang. AdaCoSeg: Adaptive shape co-segmentation with group consistency loss. In *CVPR*, 2020. 2

[77] C. Zhu, K. Xu, S. Chaudhuri, R. Yi, and H. Zhang. SCORES: Shape composition with recursive substructure priors. *ACM Trans. Graph.*, 37(6):211:1–211:14, 2018. 3

Chun-Yu Sun received his bachelor's degree in Computer Science and Technology from Xidian University in 2015. He is currently a Ph.D. student at the Institute for Advanced Study, Tsinghua University. His research interests include computer graphics and 3D Vision.

Yu-Qi Yang He received his bachelor's degree in Computer Science from the University of Science and Technology of China in 2018. He is currently a Ph.D. student at the Institute for Advanced Study, Tsinghua University. His research interests include computer graphics and 3D Vision.

Hao-Xiang Guo received his bachelor's degree in Applied Mathematics from the University of Science and Technology of China in 2017. He is currently a Ph.D. student at the Institute for Advanced Study, Tsinghua University. His research interests include 3D reconstruction, geometry processing, and shape analysis.

Peng-Shuai Wang is a researcher at Microsoft Research Asia. He received his Ph.D. degree in Computer Science and bachelor's degree in Automation both from Tsinghua University in 2018 and 2013 respectively. His research interests include computer graphics and 3D vision.

Xin Tong is a principal research manager at Microsoft Research Asia, where he leads the Internet Graphics Group. He received his Ph.D. degree from Tsinghua University in 1999. His research interests include computer graphics and computer vision, including texture synthesis, appearance modeling, light transport simulation and acquisition, 3D facial animation, and data-driven geometric processing. He has been on the editorial boards of *IEEE Transactions on Visualization and Computer graphics*, *ACM Transactions on Graphics*, and *Computer Graphics Forum*.

Yang Liu is a principal researcher at Microsoft Research Asia. He received his Ph.D. degree from the University of Hong Kong in 2008, and master's and bachelor's degrees in Computational Mathematics from the University of Science and Technology of China in 2003 and 2000. His recent research focuses on geometry processing and 3D learning. He is on the editorial boards of *IEEE Transactions on Visualization and Computer graphics* and *ACM Transactions on Graphics*.

Heung-Yeung Shum is an adjunct professor at the Institute for Advanced Study, Tsinghua University. He received his Ph.D. degree in robotics from the School of Computer Science, Carnegie Mellon University. He was the Executive Vice President of Artificial Intelligence & Research at Microsoft until March 2020. His research spans computer vision, computer graphics, pattern recognition, statistical learning, and robotics. He is a Fellow of the Institute of Electrical and Electronics Engineers and a Fellow of the Association for Computing Machinery.