Pattern Recognition Letters 179 (2024) 151–157

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Data-efficient 3D instance segmentation by transferring knowledge from synthetic scans

Xiaodong Wu, Ruiping Wang *, Xilin Chen

Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China
University of Chinese Academy of Sciences, Beijing, 100049, China

ARTICLE INFO

Editor: Lijun Yin

MSC:
41A05
41A10
65D05
65D17

Keywords:
Point cloud segmentation
Synthetic data
Domain adaptation

ABSTRACT

The 3D comprehension ability of indoor environments is critical for robots. While deep learning-based methods have improved performance, they require significant amounts of annotated training data. Nevertheless, the cost of scanning and annotating point cloud data in real scenes is high, leading to data scarcity. Consequently, there is an urgent need to investigate data-efficient methods for point cloud instance segmentation. To tackle this issue, we propose to leverage the geometric and scene context knowledge inherent in synthetic data to reduce the need for annotation on real data. Specifically, we simulate the process of human scanning and collecting point cloud data in real-world scenes and construct three large-scale synthetic point cloud datasets using synthetic scenes. The scale of these three datasets is more than ten times that of currently available real-world data. Experimental results demonstrate that the incorporation of synthetic point cloud data can increase instance segmentation performance by over **18.8** percentage points. Further, to address the problem of domain shift between synthetic and real data, we propose a target-aware pre-training method. It integrates both real and synthetic data during the pre-training process, allowing the model to learn a feature representation that can effectively generalize to downstream real data. Experiments show that our method achieved stable improvements on all three synthetic datasets. The data and code will be publicly available in the future.

1. Introduction

3D instance segmentation is an important task for indoor scene understanding. It serves as a fundamental technology for robotic perception capabilities, and as such, an increasing number of researchers are devoted to this task [1,2]. With the success of deep learning, the performance of this task has steadily improved. Although promising, the demand for huge annotated data is quite expensive. Compared with 2D image understanding tasks, collecting 3D point clouds in the real world and annotating them manually is more labor-demanding. This leads to a scarcity of real-world 3D point cloud data.

Considering the high cost of annotating 3D data, it is highly meaningful to explore how to achieve accurate 3D scene understanding with as little manual annotation as possible. There is an urgent demand for conducting research on data-efficient point cloud instance segmentation. One type of popular approach [3,4] is to utilize contrastive learning methods, which employ self-supervised loss to obtain generalizable feature representations. Another approach is to fully utilize the unlabeled data through self-training strategies [5–7]. Although these approaches can achieve certain effects, due to the scarcity of 3D data itself, their methods have limited improvements.

In this study, we propose to use external knowledge to reduce reliance on annotated real data. In recent years, there has been a significant accumulation of CAD synthetic data in both industry and academia. These synthetic data contain implicit knowledge about object shapes and scene context that is informed by human expertise. We leverage the pre-existing synthetic data accumulation to create synthetic point cloud data by mimicking the scanning and data collection process used in real-life scenarios. This allows us to obtain a large amount of annotated point cloud data at a low cost.

Collecting synthetic point cloud data mainly involves two steps: constructing synthetic scenes and scanning the synthetic scenes with a virtual camera. For synthetic scene construction, as shown in Fig. 1, we classify it into three types based on the amount of prior human knowledge involved in generating the layout: randomly generated layout, rule-based layout, and manually designed layout. There are prior works that can be utilized for these three methods, which have explored the use of synthetic data in other fields such as synthetic 2D data [8], embodied tasks [9], and SLAM [10]. For scanning the synthetic scenes, as shown in Fig. 2, we capture RGB-D frames with

* Corresponding author at: Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China.
E-mail addresses: xiaodong.wu@vipl.ict.ac.cn (X. Wu), wangruiping@ict.ac.cn (R. Wang), xlchen@ict.ac.cn (X. Chen).

https://doi.org/10.1016/j.patrec.2024.02.001
Received 24 August 2023; Received in revised form 15 December 2023; Accepted 4 February 2024
Available online 7 February 2024

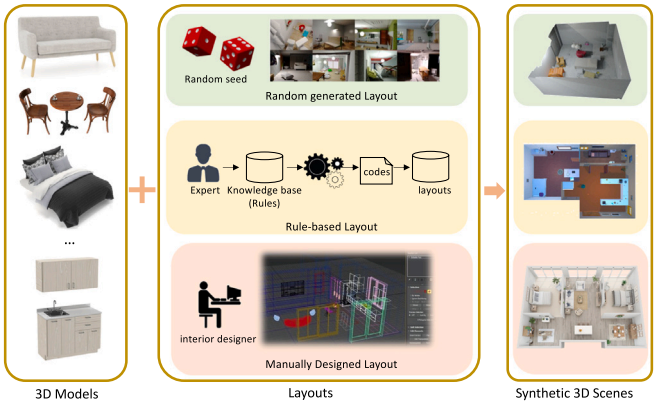

Fig. 1. A synthetic scene contains two main components, 3D object models and scene layouts. The scene layout describes the position and orientation of objects within the scene. There are three ways to generate layouts: (1) Synthetic layouts with random object poses. (2) Rule-based layouts using indoor scene priors. (3) Manually designed layouts.

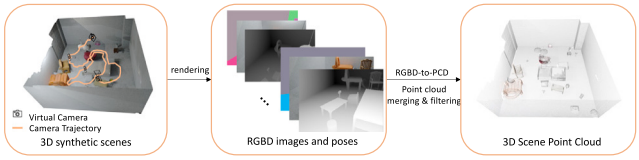

Fig. 2. Pipeline of scanning and generating point cloud from synthetic scenes.

a virtual camera and apply a 3D reconstruction algorithm to get the synthetic point cloud data. By scanning these three types of synthetic scenes, we have constructed three large-scale synthetic point cloud datasets, *ScenenNetScan*, *procTHORScan*, and *InteriorNetScan*. Table 1 shows the statistical information of the synthetic point cloud datasets. It can be seen that compared to existing point cloud datasets, our dataset has a scale of more than ten times.

Through supervised pre-training on the synthetic dataset, we can learn category-specific shape knowledge and context knowledge of the scene. The learned knowledge can be transferred to downstream tasks with only a small amount of real data labeled for fine-tuning, resulting in more accurate instance segmentation. Compared with the train-from-scratch method, using synthetic data can improve performance by more than 18 points. Furthermore, experiments show that the feature representation learned on our synthetic data exceeds that learned on real dataset, which fully demonstrates the effectiveness of the collected dataset.

The domain discrepancy between synthetic and real data can hinder the effectiveness of knowledge transfer from synthetic data. To overcome this challenge, we propose a novel target-aware pre-training approach that leverages real data during the pre-training process on synthetic data. By incorporating real data, we fully exploit its geometric and contextual knowledge to learn a feature representation that can generalize to downstream real data, thus enhancing the transferability of our model. Our approach is validated on the collected three large-scale synthetic datasets, and experimental results demonstrate its superiority over direct fine-tuning.

2. Related works

In this section, we first review the existing 3D real and synthetic data. We then introduce related works on point cloud instance segmentation and data efficiency.

Early indoor 3D scene understanding was based on RGB-D datasets, such as NYU v2 [11], which contains 1449 frames with semantic segmentation annotations, and SUN RGB-D [12], which contains 10,335 frames of RGB-D images with 2D and 3D detection annotations. These datasets provide limited 3D information, usually from a single viewpoint of the scene. Later, a group of 3D scanned scene point cloud datasets based on surface reconstruction emerged. They acquire data using RGB-D cameras or Matterport cameras in the scene and obtain mesh models through reconstruction algorithms. Due to the high cost of collecting 3D data, the amount of data obtained in real-world scenarios is usually limited. Representative works with datasets containing a small number of samples include S3DIS [13], SceneNN [14], which typically contain only one or two hundred scenes. Even datasets with a larger amount of data, such as ScanNet [15] and 3RScan [16], have only a little over a thousand scenes.

Table 1
Comparison between existing real-world point cloud datasets and our newly collected synthetic point cloud datasets.

Type	Datasets	# Scans	# Classes
Real-world	S3DIS [13]	271	13
	ScanNet [15]	1613	20
	3RScan [16]	1482	27
Synthetic	InteriorNetScan	19,951	40
	procTHORScan	36,000	40
	SceneNetScan	17,687	56

In recent years, both industry and academia have accumulated synthetic 3D data. In terms of 3D object data, ShapeNet [17] and ModelNet40 [18] are datasets of single-object CAD models. In terms of synthetic 3D scenes, academic research has developed simulation environments for agent-related embodied tasks, such as AI2THOR [19], which was designed by experts. Although it has strong interaction capabilities, the number of scenes is still limited. Subsequent work, such as procTHOR [9], solved the problem of scene scale by generating scenes using design rules. In industry, interior design companies have accumulated a large number of synthetic indoor scenes, such as InteriorNet [10] and 3D-FRONT [20]. Song et al. collect SUNCG [21] which is a manually created large-scale dataset of synthetic 3D scenes. As of now, it is no longer publicly available. There is currently a lack of scene scan point cloud data similar to ScanNet. In this work, we focus on constructing scan point cloud datasets using a data collection method similar to that used in real-world scenarios in synthetic scenes, and explore how to use these large-scale synthetic point cloud data to solve the Data Efficiency problem in real-world data.

Point cloud instance segmentation is an important foundational technology for 3D scene understanding. Hou et al. [22] use 3D convolutions to generate 3D anchor bounding box proposals and use 3D-RPN and 3D-RoI to infer object bounding box locations, class labels, and per-voxel instance masks. Yang et al. [2] propose to speed up the inferencing speed using a single-stage, anchor-free, NMS-free method. Engelmann et al. [23] propose to generate proposals by predicting object centers. PointGroup [1] proposes to cluster in both the Euclidean space and the voted center space. Recently, there are also works like MASK3D [24] and SPFormer [25] using transformers to boost the performance.

A prominent solution for achieving data-efficient point cloud segmentation involves the acquisition of proficient feature representations. PointContrast [4] and CSC [3] stand as unsupervised pre-training methods that are designed with a focus on point-level contrastive loss. Meanwhile, TWIST [6] centers its attention on the semi-supervised setting and proposes the utilization of unlabeled data to enhance model performance. Another avenue to explore is harnessing the wealth of knowledge encapsulated within synthetic data. While certain related works exist in the realm of 2D tasks [8,26], the domain of 3D indoor scene comprehension employing synthetic data remains relatively underexplored. RandomRooms [27] employs synthetic object CAD models to construct an instance-level contrastive loss framework, thereby facilitating the learning of point cloud feature representations. In this work,

we are committed to generating synthetic point clouds that resemble real point clouds and attempting to learn scene context and object shape knowledge within them. Furthermore, we embark on preliminary investigations into optimizing the utilization of synthetic data. We emphasize that the disparity between synthetic and real data domains can hinder effective knowledge transfer when employing the conventional pre-training followed by fine-tuning approach. To surmount this challenge, we introduce a novel pre-training methodology within this paper, tailored to address and mitigate domain dissimilarity issues.

3. Synthetic point cloud dataset construction

In real-world scenarios, point cloud datasets are typically collected using handheld devices to scan the environment and generate point cloud data through 3D reconstruction algorithms. This has resulted in existing 3D point cloud datasets, such as ScanNet [15] and S3DIS [13], being relatively small in scale. Furthermore, annotating point cloud data is time-consuming and costly. It is therefore promising to seek to construct synthetic scenes and virtually scan point cloud data in synthetic scenes. In this section, we first introduce synthetic scene construction in Section 3.1 and point cloud generation in Section 3.2. Then, we validate the effectiveness of the collected synthetic data in Section 3.3.

3.1. Synthetic scenes

The first step in obtaining point cloud data is to construct synthetic scenes using the 3D models. As shown in Fig. 1, there are two primary components, large-scale 3D object model data and scene layout information about object placement and pose. Object models are available in the academic research field such as ShapeNet [17] and ModelNet [18] datasets, and a large amount of data has been accumulated in industrial interior decoration and online furniture shopping platforms.

The layout of a scene determines the positions and postures of objects. It includes contextual information about the environmental context and is crucial for understanding the scene. The generation of scene layouts involves human prior knowledge about scenes. Depending on the degree of prior knowledge, as shown in Fig. 1, we can categorize scene layout construction into three types.

Randomly generated layout. A simple way to construct a scene is to randomly place objects in a room. A physically plausible scene can be obtained using a physics engine. This method has the lowest construction cost and requires minimal human prior knowledge. SceneNet RGB-D [8] is representative of this type of work, and the authors hope to use this simple scene construction method to cheaply collect 2D synthetic images with annotations.

Rule-based layout Real-life environments often adhere to specific patterns and priors in their layouts. In a household, rooms tend to follow particular distributions, and objects exhibit consistent co-occurrence relationships and relative positions. Items are positioned according to rules, such as placing refrigerators in corners and against walls, and cabinets attached to walls. By manually summarizing these scene priors, rule-based algorithms like procTHOR [9] can generate house layouts based on this knowledge. Unlike the fully random approach, this method incorporates more human-derived priors and knowledge, as seen in procTHOR, which generates synthetic environments for embodied tasks.

Manually designed layout In addition to the above two approaches, the industry has also accumulated a large amount of 3D scene data, especially in the field of interior design. Countless interior scene models have been designed by home decorators over the past few years. People use these synthesized 3D scenes as blueprints to decorate real houses. Therefore, this type of data can be considered the closest to real-world indoor scene data. InteriorNet [10] leverages the availability of professional interior designs and renders video sequences to benchmark Simultaneous Localization and Mapping (SLAM). We aim to use this type of data to generate scanned point cloud data. Compared to the first two methods of layout generation, this expert-designed approach contains the most human priors, and its design and collection costs are also the highest.

Fig. 3. Visualization of the real-world scanned point cloud dataset ScanNet and our newly collected synthetic point cloud datasets SceneNetScan, procTHORScan and InteriorNetScan.

3.2. Point cloud generation

Collecting point cloud data in indoor scenes, such as ScanNet [15], typically requires personnel to capture a video using an RGB-D camera. 3D reconstruction algorithm such as BundleFusion is applied to obtain the 3D mesh model and the point cloud of the scene. On average, collecting one ScanNet scene requires 14.9 min, while annotating one scene takes 22.3 min.

To make the synthesized point cloud data closer to the point cloud data collected from real scenes, we adopted a similar process for data collection. The process is illustrated in Fig. 2. Given a synthesized scene, we set a virtual camera to capture multiple RGB-D images and simultaneously record the camera poses. As it is a synthesized scene, we can easily obtain ground-truth 2D semantic and instance segmentation annotations. With the camera intrinsic, we can convert the RGB-D images and semantic annotations to point clouds and transform them to the world coordinate system using the camera extrinsics. Then, by downsampling the point clouds with voxelization in the world coordinate system, we can obtain the final point cloud data with annotations.

Specifically, to generate a point cloud PC_{frame} from the Depth image I_{depth} with the associated camera pose P_{cam}, we utilize the camera intrinsic matrix K and the extrinsic matrix $P_{cam} = [R \mid t]$ and convert the pixel coordinates to 3D points in the camera frame:

$$\begin{aligned} PC_{frame} &= \text{PointCloud}(I_{depth}, K, P_{cam}) \\ &= \{(X, Y, Z) \mid X = \frac{D \cdot (u - c_x)}{f_x}, Y = \frac{D \cdot (v - c_y)}{f_y}, Z = D\} \end{aligned} \tag{1}$$

$$PC_{\text{world}} = P_{\text{cam}} \cdot PC_{\text{frame}} \tag{2}$$

Here, (u, v) represents the pixel coordinates in the Depth image, D denotes the depth value, and (c_x, c_y) are the principal points, while (f_x, f_y) represent the focal lengths. Then we concatenate these point clouds into a combined point cloud and apply voxel sampling to generate the final point cloud:

$$PC_{combined} = f_{\text{downsample}}\left(\bigcup_{i=1}^{N} PC_{world_i}\right) \tag{3}$$

Building upon the existing work in synthetic scene generation, we acquired three large-scale point cloud datasets that correspond to the three layout generation methods introduced previously.

Table 2
Comparison of data-efficient instance segmentation results (AP@50) on ScanNet v2. Models are initialized from scratch or from a model pre-trained on a synthetic or real dataset. Numbers in parentheses show improvement over the baseline.

	1%	5%	10%	20%
From scratch (Baseline)	7.1	23.1	41.9	50.5
Fine-tune from 3RScan	22.4	40.4	51.6	53.7
Fine-tune from SceneNetScan	21.1 (+14.0)	37.1 (+14.0)	49.6 (+7.7)	55.0 (+4.5)
Fine-tune from procTHORScan	19.6 (+12.5)	39.6 (+16.5)	52.8 (+10.9)	55.5 (+5.0)
Fine-tune from InteriorNetScan	**24.1** (**+17.0**)	**41.9** (**+18.8**)	**54.4** (**+12.5**)	**56.7** (**+6.2**)

SceneNetScan. SceneNetScan dataset is sourced from SceneNet RGB-D [8]. The authors of SceneNet RGB-D synthesized 3D scenes by randomly placing ShapeNet [17] object models on top of SceneNet [28] 3D house models. The camera trajectories were simulated to mimic the process of handheld camera capture. Each scene produced a video sequence containing 300 frames. We fused these RGB-D video streams into scene point clouds, resulting in an indoor scene point cloud dataset that contains 56 semantic categories and 17,687 scenes. Among them, 1000 scenes were used for validation, and the remaining data were used for training.

procTHORScan. The procTHORScan dataset was obtained by capturing frames on synthesized scenes from procTHOR [9]. The authors of procTHOR designed a set of scene-setting rules to freely generate 3D scenes to facilitate research on embodied tasks. We randomly selected 20 shooting angles to capture data in each scene, and then obtained scene scan point clouds by fusing and downsampling multiple point clouds. After scanning each scene three times, we ultimately obtained a dataset containing 36,000 scans. In terms of categories, we selected and merged 40 semantic categories. Based on the common sizes of objects, we also divided them into three categories: large, medium, and small. Therefore, this dataset can support future research on 3D small object detection and segmentation.

InteriorNetScan. The authors of the InteriorNet [10] dataset utilized 3D synthetic scenes designed by interior designers to generate RGB-D video streams, which were subsequently used as benchmarks for SLAM. Compared to other datasets, it contains more human priors, and the scene settings and model details are more realistic. By converting the RGB-D image set into point clouds and fusing them, we can obtain a point cloud dataset that contains 40 categories and 19,951 scenes.

Table 1 presents the statistical information of the three datasets. It can be observed that the synthesized scene data we collected is over ten times larger than existing commonly used datasets in terms of data volume. Fig. 3 shows a comparison between our collected dataset and the real dataset ScanNet. Next, we will experimentally verify the effectiveness of synthesized data in aiding the real-scene point cloud instance segmentation task.

3.3. Validation of dataset effectiveness

Synthesized scene data can generate large amounts of data at a low cost, reducing the data demand for real scenes. Considering the cost of data collection and annotation, this is crucial for practical applications in industry and indoor scene robot understanding. To verify this, we conducted data efficiency experiments on the real-world dataset ScanNet, using {1%/5%/10%/20%} of real training data for training and evaluating the instance segmentation performance on the validation set. Considering that we mainly focus on learning 3D geometric shapes and that there are certain differences between synthesized and real data in terms of texture, we only used the position information of point clouds as feature input in our experiments.

Table 3
Instance segmentation results on the ScanNet limited annotation benchmark (200 points). Models are initialized from scratch or from a model pre-trained on a synthetic dataset (InteriorNetScan).

	AP	AP@50	AP@25
From scratch	28.9	48.8	63.1
Finetune InteriorNetScan	**33.2**	**53.3**	**68.9**

First, we conduct supervised pre-training on a large-scale synthetic dataset to fully explore the geometric structure and contextual information of the scenes. Subsequently, we utilize the learned model as initialization and fine-tune it on a small amount of annotated real data, transferring the knowledge learned from synthetic data to real data. Table 2 shows the results of our method. Compared with the baseline method that trains from scratch on limited real data only, pre-training on all three datasets can significantly improve the instance segmentation performance on real data. In particular, InteriorNetScan shows the most remarkable improvement, with an increase of **18.8** points in AP50 when using only 5% of the training data. Among the three synthetic datasets, InteriorNetScan shows the most significant improvement, followed by procTHOR and SceneNetScan. This is because InteriorNet is manually designed and contains the most human prior knowledge, making it closest to real data.

To further demonstrate the effectiveness of constructing synthetic data, we compared it with another real dataset, 3RScan [16]. We pre-train on 3RScan and fine-tune the model on a small amount of ScanNet v2 data. By comparing the results in Table 2, we found that our constructed synthetic data has surpassed the real data 3RScan.

In addition to the Limited Reconstruction benchmark, we also evaluated the ScanNet Limited Annotation (LA) benchmark. The experimental details are available in the supplementary materials. We report AP, AP@50, and AP@25 as evaluation metrics. The results of the experiment are presented in Table 3. It is evident that instance segmentation performance improved after pretraining with synthetic data.

4. Target-aware pre-training for transferring knowledge from synthetic to real

The method introduced in the previous section is directly pre-trained on the synthetic dataset and fine-tuned on the real dataset. We consider synthetic data as the source domain and real data as the target domain. The problem is that pre-training on the source domain alone ignores the target domain's data distribution and domain differences, leading to a suboptimal performance on real data.

In this section, we propose a pre-training method that incorporates the unlabeled target domain data in the pre-training process. It enables the model to acquire the geometric structures and scene contexts of point clouds from both the source and target domains and facilitates the transferability of the pre-trained model to the target domain task. The main insight is to allow the network to capture the data distribution of the target domain during the pre-training process. We need to involve the target domain data in the pre-training process and design a loss function to guide the network's learning. To achieve this, we generate pseudo-labels for the unlabeled target domain data in the label space of the source domain and use this as supervised information for pseudo-training.

4.1. Problem definition

The goal of this work is to achieve accurate instance segmentation on a real-world collected point cloud dataset $D_t = \{D_t^u, D_t^l\}$, where only a small fraction of the data is labeled, D_t^u and D_t^l represent the unlabeled and labeled part separately. To address the challenge of scarce annotations, we propose a method that leverages a large-scale, inexpensive, and readily available synthetic dataset $D_s = \{(\boldsymbol{P}_i^s, \boldsymbol{Y}_i^s)\}_{i=1}^{N_s}$

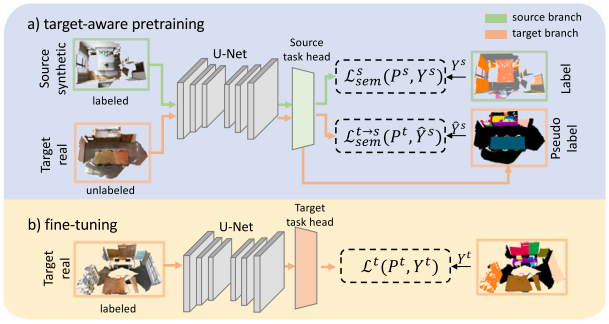

Fig. 4. The training process of our proposed method is composed of two phases: (a) target-aware pertaining, (b) fine-tuning.

to fully learn the geometric structure of point clouds. The model trained on both the labeled synthetic dataset $\mathcal{D}_s$ and unlabeled real dataset $\mathcal{D}_t^u$ is used to initialize the model parameters and then fine-tuned on the real data. Subscripts s and t denote the source domain and target domain, respectively.

4.2. Method

The overall framework is shown in Fig. 4 and it is composed of two phrases, i.e, target-aware pre-training and fine-tuning. During the target-aware pre-training phase, we leverage both labeled data from the source domain and unlabeled data from the target domain as input to learn knowledge on both domains. This approach not only helps the model learn the geometric structure and scene context of point clouds in the source domain but also improves its generalization performance in the target domain. To fully utilize the information from both labeled and unlabeled data, we design two loss functions for network learning on source and target domain data, respectively. In the source domain, we use labeled data for supervised training to facilitate the learning of knowledge in the source domain. Formally, for a point cloud $\boldsymbol{P}_i^s$ with N points, the model produces semantic scores $\boldsymbol{SC}_i = \{sc_1, \dots, sc_N\} \in \mathbb{R}^{N \times N^s_{class}}$. We use the cross-entropy $\mathcal{L}^s_{sem}$ for supervised training:

$$\mathcal{L}^s_{sem} = \frac{1}{N_s} \sum_{i=1}^{N_s} CE(\boldsymbol{SC}_i, \boldsymbol{Y}_i^s) \tag{4}$$

where $CE(\cdot,\cdot)$ is the cross-entropy loss function. For the target domain, we adopt a self-training learning approach to generate pseudo labels of the source domain task and use them for supervised training to facilitate the network learning of feature representation and data distribution in target domain point cloud data. We first generate pseudo label $\hat{\boldsymbol{Y}}_i^s = \{\hat{\boldsymbol{Y}}_{i,j}^s\}_{j=1}^N$ of source task for unlabeled target point cloud $\boldsymbol{P}_i^t$ as follows:

$$\hat{\boldsymbol{Y}}_{i,j}^s = \begin{cases} arg\,max(\boldsymbol{SC}_{i,j}) & max(\boldsymbol{SC}_{i,j}) > T \\ -1 & \text{otherwise} \end{cases} \tag{5}$$

where j is the index of point and T is a pre-defined confidence threshold. The point is ignored if the probability is smaller than the threshold and is assigned with the label -1. Then, pseudo-loss $\mathcal{L}^{t\to s}_{sem}$ is used to supervise the training on target data:

$$\mathcal{L}^{t\to s}_{sem} = \frac{1}{N_t} \sum_{i=1}^{N_t} CE(\boldsymbol{SC}_i, \hat{\boldsymbol{Y}}_i^s) \tag{6}$$

The final pre-training loss is calculated as $\mathcal{L} = \mathcal{L}^s_{sem} + \lambda \mathcal{L}^{t\to s}_{sem}$ and λ is a hyper-parameter to tune the influence factor of the target domain. After the pre-training process, the model can be fine-tuned on the limited labeled target point cloud data to yield a well-performing instance segmentation model.

4.3. Experimental settings

4.3.1. Datasets and metrics

The three synthetic datasets SceneNetScan, procTHORScan and InteriorNetScan collected in this work are used as the source domain datasets. We use the ScanNet v2 [15] dataset as the target real dataset. It contains 1613 scans and 20 classes. 10% of the entire 1201 scenes in the training set for training and evaluate on the validation set. AP_{50} denotes the average precision with IoU threshold 50% and is used as the evaluation metric of instance segmentation performance.

4.3.2. Implementation details

We use PointGroup [1] as the point cloud instance segmentation method. The voxel size of the point cloud is set as 0.02 m. The model is trained with an initial learning rate of 0.001 and decays with the OneCycle policy [29]. AdamW [30] is used as the solver. In the target-aware pre-training process, the hyper-parameter λ is set as 0 in the first 256 epochs and set as 1 in the following 128 epochs. In the fine-tuning process, we initialize the model parameters with the pre-trained model and train it for 512 epochs.

4.4. Evaluation results

We compare our method with a baseline approach which is solely pre-trained on the synthetic dataset and is transferred to the downstream real dataset by fine-tuning. Table 4 shows that our method outperforms it on all three synthetic datasets.

It is noteworthy that our method exhibits significant improvements on categories with large differences between synthetic and real domains. For instance, the *shower curtain* category, which is less frequently present in the synthetic dataset, achieved performance gains of 14.2, 4.0, and 10.1 over the baseline method on the SceneNetScan, procTHORScan, and InteriorNetScan datasets, respectively. In contrast, for categories with smaller differences between synthetic and real domains, such as *chair*, *toilet*, and *door*, the adoption of target-aware pre-training has little impact. This indicates that our method not only learns rich feature representations on the source domain during pre-training, but also fully utilizes the data from the target domain to facilitate the model's transferability to the target domain, thus allowing for efficient and effective learning from limited labeled real data. We also find that introducing unsupervised domain adaptation methods such as Mix3D [31] during the pre-training process does not perform well. Its AP@50 is 2.2 points lower than our approach (53.2 vs. 55.4).

We present in the first row of Table 4 the performance of the model trained solely on 100% real data. As shown, our method combined with a large synthetic dataset has enabled the 10% labeled data to achieve performance similar to that of using 100% labeled data. This is highly beneficial for practical applications to reduce expensive annotation costs.

Our proposed method outperformed both the training-from-scratch and fine-tuning approaches, yielding superior instance segmentation results. To further demonstrate the effectiveness of our method, we compare it with existing methods on ScanNet v2 validation set with various ratios of labeled data. Table 5 shows that our method outperforms all existing methods by a large margin. This strongly indicates that learning efficient and easily transferable feature representations from synthetic data is an effective solution for addressing the scarcity of target annotated data. In Fig. 5, we show the visualization results of our method on ScanNet v2.

5. Conclusion

In this work, we investigate the possibility of constructing point cloud datasets using synthetic scenes. We have collected three

Table 4
3D instance segmentation results on ScanNet v2 validation set. Limited data (10%) is used for training. We first pre-train on three synthetic datasets using the baseline method and our target-aware pre-training method and report their performances on the real dataset.

Method	Synthetic dataset	Real dataset	AP_{50}	bath.	bed.	bkshf.	cab.	chair.	cntr.	curt.	desk.	door.	ofrun.	pic.	fridg.	showr.	Sink	Sofa	Table	Toilet	wind.
Supervised training	None	ScanNet (100%)	59.9	83.9	74.9	45.4	53.5	86.7	29.7	50.6	46.1	41.9	58.2	40.1	49.6	68.9	70.8	62.4	71.9	98.3	44.8
Supervised training	None	ScanNet (10%)	42.6	76.9	70.2	51.1	29.0	82.8	13.5	36.6	21.0	25.2	41.7	0.0	1.8	57.8	23.2	64.7	55.0	89.0	27.0
Fine-tuning	SceneNetScan	ScanNet (10%)	49.6	67.4	72.7	50.9	39.3	87.9	22.8	41.7	20.0	34.0	47.5	25.5	17.7	62.9	48.5	72.9	54.6	94.7	31.9
Ours	SceneNetScan	ScanNet (10%)	**52.7**	80.5	75.8	62.2	36.7	87.2	24.6	40.4	26.9	32.7	49.3	23.4	27.8	77.1	49.8	69.1	52.0	94.4	37.9
Fine-tuning	procTHORScan	ScanNet (10%)	52.8	77.4	74.1	47.6	43.5	85.2	22.2	36.5	39.5	35.1	43.5	31.5	30.5	68.7	62.0	64.9	62.9	91.2	33.6
Ours	procTHORScan	ScanNet (10%)	**53.0**	77.3	71.2	46.9	43.6	86.4	22.4	42.5	34.2	35.6	43.8	33.7	30.1	72.7	51.8	78.3	58.8	92.9	32.0
Fine-tuning	InteriorNetScan	ScanNet (10%)	54.4	86.9	65.1	49.1	46.2	85.2	28.7	46.1	33.7	42.5	44.7	37.8	32.4	65.2	57.7	63.1	56.1	96.0	43.3
Ours	InteriorNetScan	ScanNet (10%)	**55.4**	87.0	69.6	54.5	45.1	86.6	28.5	37.9	41.4	41.3	40.0	40.1	29.3	75.3	55.3	64.5	59.9	99.8	41.0

Table 5
Comparison of data-efficient instance segmentation results (AP@50) with various ratios of labeled data on ScanNet v2 validation set. Our method based on InteriorNetScan outperforms all existing methods.

Method	1%	5%	10%	20%
PointContrast [4]	12.5	35.4	43.9	49.5
CSC [3]	13.0	36.7	45.0	50.3
TWIST [6]	17.1	44.1	49.7	52.9
Ours	**25.3**	**45.1**	**55.4**	**57.0**

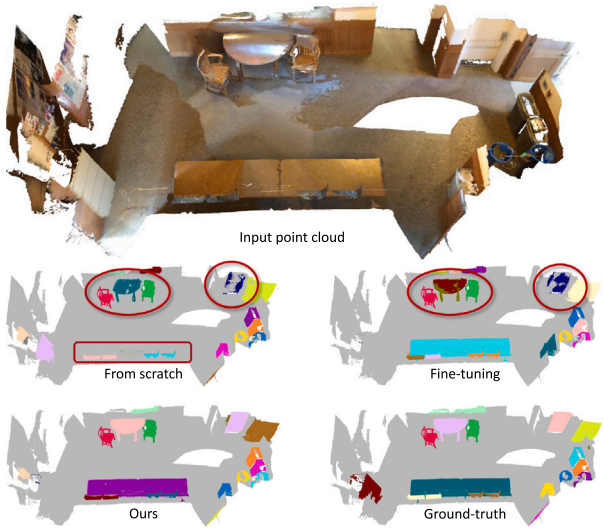

Fig. 5. Comparison of qualitative instance segmentation results on the ScanNet v2 dataset. Different colors represent separate instances. Areas circled in red show poorer predictions for the compared methods.

large-scale synthetic point cloud datasets using different scene layout generation methods, which are more than ten times larger than existing datasets. The learned feature representations from these datasets can be efficiently applied to real point cloud tasks, consequently decreasing the need for actual annotated data. Considering the domain differences between real and synthetic data, we propose a target domain-aware pre-training method that fully explores the geometric shape knowledge of synthetic and real data during pre-training and performs better when transferred to real data. Furthermore, the synthetic datasets we collected may also be used for other tasks, such as self-supervised contrastive learning and sim2real research in the point cloud domain. Additionally, multiple synthetic datasets can serve as benchmarks for multi-domain knowledge transfer of point cloud, promoting research and practical applications in the field.

CRediT authorship contribution statement

Xiaodong Wu: Methodology, Writing – original draft, Conceptualization, Formal analysis, Validation. **Ruiping Wang:** Conceptualization, Funding acquisition, Methodology, Supervision, Writing – review & editing. **Xilin Chen:** Conceptualization, Funding acquisition, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is partially supported by National Key R&D Program of China No. 2021ZD0111901, and Natural Science Foundation of China under contracts Nos. U21B2025, U19B2036.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.patrec.2024.02.001.

References

[1] L. Jiang, H. Zhao, S. Shi, S. Liu, C. Fu, J. Jia, PointGroup: Dual-set point grouping for 3D instance segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 4866–4875.
[2] B. Yang, J. Wang, R. Clark, Q. Hu, S. Wang, A. Markham, N. Trigoni, Learning object bounding boxes for 3D instance segmentation on point clouds, in: Advances in Neural Information Processing Systems, NeurIPS, 2019, pp. 6737–6746.
[3] J. Hou, B. Graham, M. Nießner, S. Xie, Exploring data-efficient 3D scene understanding with contrastive scene contexts, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 15587–15597.
[4] S. Xie, J. Gu, D. Guo, C.R. Qi, L. Guibas, O. Litany, Pointcontrast: Unsupervised pre-training for 3d point cloud understanding, in: European Conference on Computer Vision, ECCV, Springer, 2020, pp. 574–591.
[5] D.-H. Lee, et al., Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, in: Workshop on Challenges in Representation Learning, ICML, Vol. 3, 2013, p. 896.
[6] R. Chu, X. Ye, Z. Liu, X. Tan, X. Qi, C. Fu, J. Jia, TWIST: two-way inter-label self-training for semi-supervised 3D instance segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 1090–1099.
[7] Y. Chen, M. Nießner, A. Dai, 4DContrast: Contrastive learning with dynamic correspondences for 3D scene understanding, in: European Conference on Computer Vision, ECCV, Vol. 13692, 2022, pp. 543–560.
[8] J. McCormac, A. Handa, S. Leutenegger, A.J. Davison, Scenenet rgb-d: Can 5 m synthetic images beat generic imagenet pre-training on indoor segmentation? in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2678–2687.

[9] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, K. Ehsani, J. Salvador, W. Han, E. Kolve, A. Kembhavi, R. Mottaghi, ProcTHOR: Large-scale embodied AI using procedural generation, Adv. Neural Inf. Process. Syst. 35 (2022) 5982–5994.
[10] W. Li, S. Saeedi, J. McCormac, R. Clark, D. Tzoumanikas, Q. Ye, Y. Huang, R. Tang, S. Leutenegger, InteriorNet: Mega-scale multi-sensor photo-realistic indoor scenes dataset, in: British Machine Vision Conference, BMVC, 2018, p. 77.
[11] N. Silberman, R. Fergus, Indoor scene segmentation using a structured light sensor, in: IEEE International Conference on Computer Vision Workshops, ICCV Workshops, IEEE Computer Society, 2011, pp. 601–608.
[12] S. Song, S.P. Lichtenberg, J. Xiao, SUN RGB-D: a RGB-D scene understanding benchmark suite, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015, pp. 567–576.
[13] I. Armeni, O. Sener, A.R. Zamir, H. Jiang, I.K. Brilakis, M. Fischer, S. Savarese, 3D semantic parsing of large-scale indoor spaces, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 1534–1543.
[14] B. Hua, Q. Pham, D.T. Nguyen, M. Tran, L. Yu, S. Yeung, SceneNN: A scene meshes dataset with annotations, in: 3DV 2016, IEEE Computer Society, 2016, pp. 92–101.
[15] A. Dai, A.X. Chang, M. Savva, M. Halber, T.A. Funkhouser, M.N. ner, ScanNet: Richly-annotated 3D reconstructions of indoor scenes, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 2432–2443.
[16] J. Wald, A. Avetisyan, N. Navab, F. Tombari, M. Nießner, RIO: 3D object instance re-localization in changing indoor environments, in: IEEE International Conference on Computer Vision, ICCV, 2019, pp. 7657–7666.
[17] A.X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al., Shapenet: An information-rich 3d model repository, 2015, arXiv preprint arXiv:1512.03012.
[18] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3D ShapeNets: A deep representation for volumetric shapes, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015, pp. 1912–1920.
[19] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu, et al., Ai2-thor: An interactive 3d environment for visual ai, 2017, arXiv preprint arXiv:1712.05474.
[20] H. Fu, B. Cai, L. Gao, L. Zhang, J. Wang, C. Li, Q. Zeng, C. Sun, R. Jia, B. Zhao, H. Zhang, 3D-FRONT: 3D furnished rooms with layouts and semantics, in: IEEE International Conference on Computer Vision, ICCV, 2021, pp. 10913–10922.
[21] S. Song, F. Yu, A. Zeng, A.X. Chang, M. Savva, T.A. Funkhouser, Semantic scene completion from a single depth image, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 190–198.
[22] J. Hou, A. Dai, M. Nießner, 3D-SIS: 3D semantic instance segmentation of RGB-D scans, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 4421–4430.
[23] F. Engelmann, M. Bokeloh, A. Fathi, B. Leibe, M. Nießner, 3D-MPA: Multi-proposal aggregation for 3D semantic instance segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 9028–9037.
[24] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, B. Leibe, Mask3D: Mask transformer for 3D semantic instance segmentation, in: IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2023, pp. 8216–8223.
[25] J. Sun, C. Qing, J. Tan, X. Xu, Superpoint transformer for 3D scene instance segmentation, in: AAAI Conference on Artificial Intelligence, AAAI, 2023, pp. 2393–2401.
[26] F. Poucin, A. Kraus, M. Simon, Boosting instance segmentation with synthetic data: A study to overcome the limits of real world data sets, in: IEEE International Conference on Computer Vision Workshops, ICCV Workshops, IEEE, 2021, pp. 945–953.
[27] Y. Rao, B. Liu, Y. Wei, J. Lu, C. Hsieh, J. Zhou, RandomRooms: Unsupervised pre-training from synthetic shapes and randomized layouts for 3D object detection, in: IEEE International Conference on Computer Vision, ICCV, 2021, pp. 3263–3272.
[28] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, R. Cipolla, Understanding real world indoor scenes with synthetic data, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 4077–4085.
[29] L.N. Smith, N. Topin, Super-convergence: Very fast training of neural networks using large learning rates, in: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, Vol. 11006, 2019, pp. 369–386.
[30] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: International Conference on Learning Representations, ICLR, 2019.
[31] A. Nekrasov, J. Schult, O. Litany, B. Leibe, F. Engelmann, Mix3D: Out-of-context data augmentation for 3D scenes, in: International Conference on 3D Vision, 2021, pp. 116–125.