REFERRING SELF-SUPERVISED LEARNING ON 3D POINT CLOUD

Anonymous authors
Paper under double-blind review

ABSTRACT

After observing a type of object, we humans could easily recognize similar objects on an unseen scene. However, such generalization ability for the neural network remains not fully explored in current researches. In this paper, we study a new problem named Referring Self-supervised Learning (RSL) on 3D scene understanding: Given the 3D synthetic models with labels and the unlabeled 3D real scene scans, our goal is to distinguish the identical semantic objects on an unseen scene according to the referring synthetic 3D models. Unlike current tasks, the purpose of RSL is to study how to transfer the neural network's knowledge from the 3D models to unseen 3D scenes, where the main challenge is solving the cross-scene -domain and -task gap between the referring synthetic model and real unseen scene. To this end, we propose a simple yet effective self-supervised framework to perform two alignment operations. First, *physical alignment* aims to make the referring models match the scene with data processing techniques, and then *convex-hull regularized feature alignment* introduces learnable prototypes to project the point features of referring models to a convex hull space, where the feature acts as a convex combination of the learned prototypes (for both referring model and real scene) and this regularization eases the alignment. Experiments show that our method achieves the average mAP of 55.32% on the ScanNet dataset by referring only to the synthetic models from the ModelNet dataset. Furthermore, it can be regarded as a pretext task to improve the performance of the downstream tasks in 3D scene understanding.

1 INTRODUCTION

The human perception system has a powerful generalization ability across different scenes, domains and tasks. For example, after observing many multi-category objects, humans conclude the typical characters of the identical semantic objects and the unique characters between different categories. Therefore, they could easily distinguish the identical semantic object on an unseen scene, no matter where the object is. However, such cross-scene and cross-domain generalization ability of the neural network remains under-explored.

Most current works for the neural network are task-, scene- and domain-specific, which indicate that a specific neural network is trained for a specific task with a corresponding training dataset. Moreover, the network performs the cross-scene, cross-domain, and cross-task adaptation needs massive data labelling and computational resources. Based on this consideration, this paper investigates a new problem on the neural network's cross-scene and cross-domain generalization ability for 3D scene understanding. Given 3D referring models with labels and the unlabeled 3D scenes, our goal is to distinguish the identical semantic objects on an unseen scene according to the referring 3D models. We name it Referring Self-supervised Learning (RSL). The referring models with class tags are synthetic and easily accessible, and the 3D scenes are unlabeled scans acquired from the real world. The purpose is to study how to perform the effective and efficient cross-scene cross-domain and cross-task knowledge transferring like humans on a 3D world.

Unlike existing works, the challenging issues of the proposed Referring Self-supervised Learning are the cross-scene, cross-domain and cross-task gaps. First, it differs naturally from supervised learning (Ouaknine et al. (2021); Chen et al. (2021); Zhu et al. (2021); Li et al. (2021)), which is task-specific and often limits to very few scenarios. The straightforward idea is to utilize the unsupervised domain adaptation (Saenko et al. (2010); Wang et al. (2020)), which transfers the knowledge from the source domain to the target domain where target labels are not necessary, or few-shot/semi-supervised learning (Yu et al. (2020); Zhao et al. (2021)) via using a small proportion of labelled data. However, these studies are task-specific (from classification to classification or segmentation to segmentation) and scene-specific, while RSL emphasizes the cross-scene, cross-domain and cross-task generalization ability of the neural network, which is under-explored in these works.

In this paper, we formulate RSL on point cloud and instantiate it with the point-wise classification problem. The main challenges are the cross-domain (synthetic to real) cross-scene (single model to the indoor scene) and cross-task (classification to segmentation) gap. Specifically, unlike the referring models are independent and complete, the objects in a scene are nearby other objects and are often partially observed. They also vary in geometric properties, where the objects in a real scan are irregular and noisy due to the limitation of scanning equipment while the synthetic models are clean. To this end, we propose a simple yet effective framework to handle the domain gaps, where it mainly includes two alignment operations, *i.e*, physical alignment and convex-hull regularized feature alignment.

Specifically, we first design a series of data processing approaches to perform the physical alignment between the synthetic models and the object in a real scene, including rotation, scaling, cropping, and mix up with other models and the scene. Besides, the huge distribution difference between the synthetic model and the real-world scene makes the alignment hard. A natural alternative is to impose restrictions on these distributions to ease the alignment. Inspired by the convex hull theory (Rockafellar (2015)), *i.e*, any convex combinations of the points must hold and restrain in the convex hull, we propose a novel module to project the point features into the convex hull space that regularizes these distributions. The basic idea is to set a group of learnable prototypes as the support points to formulate a convex hull. These prototypes are designed to indicate the base properties of 3D models according to the objective function, where the convex hull is a closure subspace surrounded by the learned prototypes. When inferring an unseen scene, the point features are projected to the convex hull by a convex combination of prototypes. In this way, the convex hull regularized feature representation has a better generalization ability to recognize the target objects when inferring an unseen scene.

We conduct the experiments on ModelNet (Wu et al. (2015)) and ScanNet (Dai et al. (2017)), where ModelNet provides the synthetic model for referring, and the scene scans in ScanNet are for evaluation. Our method achieves the average mAP of 55.32% o ScanNet dataset without any manually annotated scans. Furthermore, RSL can be a pretext task to pre-training the network. The proposed method has significantly improved the existing methods with different proportions of labelled data, demonstrating a promising way for representation learning.

The contributions of our work are as follows.

- Inspired by the human perception system, we formulate and investigate a new problem about the networks' cross-domain cross-scene and cross-task generalization ability.
- We propose a novel framework to perform the alignment between the synthetic models and the objects in an unseen scene via physical space and convex-hull regularization.
- Our method achieves promising results to infer the interesting objects on unseen scenes.
- Our work indicates a feasible way for representation learning on point cloud to improve the downstream tasks.

2 Related Work

Deep Learning on Point Cloud Point cloud, as a 3D data representation, has been used extensively in various 3D related tasks, such as shape classification (Liu et al. (2021)), 3D segmentation (Ouaknine et al. (2021); Chen et al. (2021); Zhu et al. (2021)), 3D detection (Li et al. (2021)) and registration (Lu et al. (2021); Zeng et al. (2021)). Due to its unordered nature, the network for processing the point cloud is less mature than 2D images. Various network architectures have been proposed for learning with point clouds. They mainly focus on designing mechanisms for aggregating neighbourhood information. PointNet based methods (Qi et al. (2017a;b); Wu et al. (2019)) mainly apply multi-layer perception networks directly on the coordinates of input point cloud for feature extraction and cropping operation is usually employed for local feature aggregation. Continuous convolutional networks (Wang et al. (2018); Yang et al. (2021)) aim at exploring the underlying continuous structures represented with discrete points and define convolution kernels in continuous spaces. Sparse convolution-based methods (Graham (2015); Su et al. (2018); Choy et al. (2019)) mainly represent the 3D point cloud with sparse rectangles or permutohedral lattice and define 3D convolution kernels accordingly. As an extension of standard 2D convolution, various architectures designed for 2D convolution can also be employed with 3D sparse convolution. In our work, we use the MinkowskiNet (Choy et al. (2019)), a kind of sparse convolutional network, with U-Net like architecture as the backbone for learning point cloud representations.

Transfer Learning in 3D Transfer learning has been widely employed in various deep learning-based tasks. Generally, transfer learning can be roughly classified into three categories including pre-training (Yosinski et al. (2014); Mahajan et al. (2018)), domain adaptation (Saenko et al. (2010); Wang et al. (2020)) and few-shot learning (Yu et al. (2020); Zhao et al. (2021)). The main purpose of using transfer learning is to improve the generalization and stability of neural networks. It is especially important when the neural networks are trained with limited labelled data. In 3D scenarios, transfer learning becomes much more important due to the difficulty of acquiring 3D labelled data. PointContrast (Xie et al. (2020)) extends contrastive learning technique for learning point-wise representations that are beneficial to various downstream tasks. (Hou et al. (2021)) integrates the spatial scene contexts into a contrastive training paradigm and achieved considerable improvements on supervised and weakly supervised downstream tasks. In this paper, we instantiate the referring self-supervised learning on point cloud and transfer object-level classification labels from 3D synthetic models to real scanned scenes for scene segmentation tasks.

3D Data Augmentation Data augmentation, as a fundamental way for enlarging the quantity and diversity of training datasets, plays an important role in various deep learning tasks. It is especially important in the 3D deep learning scenario, which is notoriously data hungry. Simple data augmentation schemes like random rotation, translation, jittering, scaling, and cropping are commonly used in point cloud deep learning methods. Such simple techniques can usually achieve much more performance gain compared with complex network architecture designs. Recently, several attempts have been made on designing new 3D data augmentation schemes and studying 3D data augmentation techniques in systematic ways. PointAugment (Li et al. (2020)) proposes a learnable point cloud augmentation module to make the augmented data distribution better fit with the classifier, and the augmentation module is trained in an adversarial way. PointMixup (Chen et al. (2020)) extends Mixup (Zhang et al. (2017)) scheme from 2D image to 3D point cloud, it augments the data by interpolating between data samples. PointCutMix (Zhang et al. (2021)) further extend Mixup strategy and perform mixup on part level. In our work, we propose a novel Mixup strategy for referring self-supervised learning on 3D point cloud. Unlike most existing works, which mainly use data augmentation to improve the robustness and generalization of the network, we in addition use such schemes to align the 3D referring models to the corresponding objects in the real scanned scenes.

Memory Networks The Prototype-based Memory network has been applied to various problems. NTM (Graves et al. (2014)) introduces an attention-based memory module to improve the generalization ability of the network. Gong et al. (Gong et al. (2019)) adopt a memory augmented network to detect the anomaly. Prototypical Network (Snell et al. (2017)) utilize category-specific memories for few-shot classification. Liu et al. (Liu et al. (2019)) and He et al. (He et al. (2020)) solve the long-tail issue with the prototypes. In this paper, we adopt the learnable prototypes for domain alignment via mapping feature to closure and compact feature space.

3 Referring Self-supervised Learning

Problem Definition Given a set of 3D referring models with labels $\{(M_i, G_i), i = 1, 2, 3, ..., N\}$ and a bunch of unlabeled 3D scene scans $\{S_j, j = 1, 2, 3, ..., M\}$, our goal is to distinguish the identical semantic objects on a scan. We formulate it as the referring self-supervised learning and instantiate it as a point-wise classification problem, *i.e*, producing the possibility of each point that belongs to specific classes. If directly training the referring models for point-wise classification and

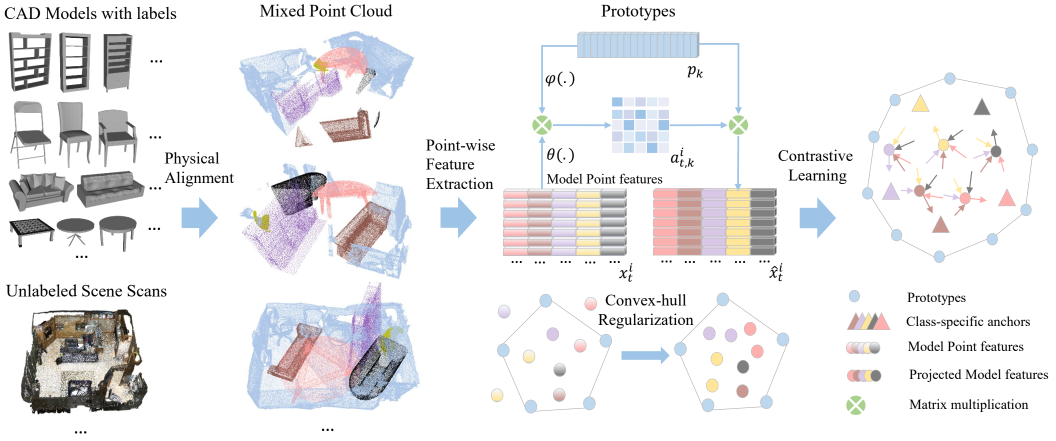

Figure 1: The framework of our method. Firstly, the referring models are aligned to the objects in a real scene scan via physical data alignment, including random rotation, scaling, cropping, and mixing up with the scene. Secondly, we extract the point-wise feature from the mixed point cloud and map the point features into a convex hull space surrounded by a group of learned prototypes. In the end, the aligned point features are clustering with the class-specific anchors in metric space.

then inferring on an unseen scan, the performance is unacceptable (Table 1) because there are huge gaps in the point feature between the 3D referring models and the identical semantic objects in an unseen scene scan.

Approach Overview As illustrated in Figure 1, our method handles the domain gaps from two aspects. Firstly, we physically align the referring models close to the objects in a real scene scan using data processing techniques. Secondly, a convex-hull regularized feature alignment is proposed, in which we map the point features into a unified convex hull space surrounded by a group of learned prototypes. In the end, we perform contrastive learning on the mapping features. In what follows, we will present these components in detail.

3.1 Physical Data Alignment

Physical-level alignment is an intuitive and straightforward option to make the alignment easier. A series of data processing approaches are introduced to cope with the referring models to align them to the objects in a real scene scan, including **1)** Point sampling from the computer-aided design (CAD) models, **2)** scaling, rotation, cropping, and **3)** mix up with the scene.

Point Sampling CAD models are presented in Mesh format that consists of the vertexes and faces. To unify the data format, we transfer the model mesh to a uniform point cloud by Poisson Disk Sampling (Yuksel (2015)), and ensure the density closer to the scene scan.

Scaling, Rotation and Cropping We scale the size of the referring models to match the object in a real scene scan for consistent local feature extraction. Besides, random rotation transformation is also applied to capture the visual diversity of an object. Finally, considering that the object in a scene scan is always partial observed, a random cropping strategy is designed to simulate this scenario. Specifically, we first randomly sample 2∼5 points from the model as anchor points and then cluster all points based on their Euclidean distance to the anchor points. When training the model, one of a cluster will be randomly filtered.

Mix Up with Scene Scan Unlike the referring model, the point feature of the target object in a real scene scan is always affected by the surrounding object. Therefore, we take some unlabeled scans to mix up with the referring models to alleviate the adverse effect. Specifically, we randomly place the referring models into the scene floor, with or without filtering the compacted scan points.

3.2 Convex-hull Regularized Feature Alignment

Since the network is only trained on the referring models, the feature space is typically out-of-the-distribution to the object in a scene scan, leading to low inferring accuracy on an unseen scene. Inspired by the convex hull theory, we propose a novel module to project the point features into a convex hull space via the learnable prototypes. In the following, we revisit the convex hull and describe how prototypes are used for feature projection.

Revisiting Convex Hull Theory Convex hull is a fundamental concept in computational geometry. It is defined as the set of all convex combinations of points, where the convex combination is a linear combination where all coefficients are non-negative and sum to 1. By definition, if a point is presented as a convex combination of the points, it must remain in the convex hull. In this way, the distributions from two different domains can be constrained and easier for alignment. Therefore, we aim to project all features into a convex hull to ease the alignment difficulty.

Formulation We set a group of learnable prototypes $\{p_k\}_{k=1}^{K}$ with $p_k \in \mathbb{R}^D$ and $K > D$, where K denotes the number of prototypes, and D is the dimension of a prototype. Note that prototypes are directly learned from the referring models and updated according to the task objective function. Given the point features $\{x_t^i\}_{t=1}^{T}$ with $x_t^i \in \mathbb{R}^D$ extracted by the encoder E from the i-th referring model with T points. The corresponding mapping feature $\{\hat{x}_t^i\}_{t=1}^{T}$ is obtained by the following function.

$$\hat{x}_t^i = \sum_{k=1}^{K} a_{t,k}^i * p_k, \sum_{k=1}^{K} a_{t,k}^i = 1, \tag{1}$$

where the function $a_{t,k}^i$ serves as the coefficient to the corresponding prototype, defined as follows.

$$a_{t,k}^i = \exp(\lambda * d(\theta(x_t^i), \varphi(p_k)))/\Gamma, \Gamma = \sum_{k=1}^{K} \exp(\lambda * d(\theta(x_t^i), \varphi(p_k))), \tag{2}$$

where $d(\cdot)$ measures the similarity between the point feature and the k-th prototype. We utilize dot product operation in this work. $\theta(\cdot)$ and $\varphi(\cdot)$ denote the key and the query function (Vaswani et al. (2017)), respectively. λ is the inversed temperature term (Chorowski et al. (2015)).

Essentially, The feature embedding $\hat{x}_t$ is a convex combination of the prototypes, and the coefficient $a_{t,k}^i > 0$. Therefore, $\hat{x}_t$ is mapped into a convex $\mathbb{W} \subset \mathbb{R}^D$, where $\mathbb{W}$ is a closure and compact metric space that surrounded by the learned prototypes.

In the inferring phase, the point feature $x \in \mathbb{R}^D$ from an unseen scan first accesses the most relevant prototypes to obtain the coefficients. Then, it is transferred to a mapping feature $\hat{x} \in \mathbb{W}$ which is the convex combination of the prototypes. In this way, the feature space of the referring models and the objects in an unseen scene are projected to the unified subspace $\mathbb{W}$, leading to the network a better generalization ability.

3.3 Contrastive learning on metric space

As all point features are projected to the subspace $\mathbb{W}$, we cluster these points to ensure they are inner-class compact and inter-class distinguishable. We set the class-specific anchors $\{h_c\}_{c=1}^{C}$ with $h_c \in \mathbb{R}^D$ to indicate the clustering centers in metric space, which include $C-1$ foreground classes and one background class. Given the point features $\{\hat{x}_t^i\}_{i=1,t=1}^{N,T}$ from N referring models$\{M_i\}_{i=1}^{N}$, we pull in whose points to the corresponding anchor while pushing away from the rest of anchors according to their semantic labels $\{G_i\}_{i=1}^{N}$. Therefore. The points are compact inner class and distinguishable inter classes in the metric space $\mathbb{W}$. For simplicity, we adopt Cross-Entropy loss in this paper. Therefore, the objective function $\mathcal{L}$ is as follows.

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{t=1}^{T} d(\hat{x}_t^i, h_{G_i}) + \sum_{i=1}^{N}\sum_{t=1}^{T} \log(\sum_{c=1}^{C} \exp(d(\hat{x}_t^i, h_c))). \tag{3}$$

When inferring an unseen scene scan S_j with the point features $\{\hat{x}_t^j\}_{i=1}^T$, the possibility distribution of the t-th point that belongs to each class is determined by the similarities between the point feature $\hat{x}_t^j$ and the class-specific anchors $\{h_c\}_{c=1}^C$.

$$l_t^c = \exp(d(\hat{x}_t^j, h_c))/\gamma, \gamma = \sum_{c=1}^{C} \exp(d(\hat{x}_t^j, h_c)), \tag{4}$$

where l_t^c is the possibility that the t-th point belongs to the c-th class, γ is a normalization term.

4 Experiments

4.1 Dataset

We conduct the experiments on ModelNet and ScanNet, where ModelNet provide the synthetic model for referring, and the scene scans in ScanNet are for evaluation.

ModelNet ModelNet40 (Wu et al. (2015)) is a comprehensive clean collection of 3D CAD models for objects, composed of 9843 training models and 2468 testing models in 40 classes. We transfer the model mesh to 8196 uniform points by Poisson disk sampling (Yuksel (2015)). There are 11 identical classes to the ScanNet dataset, including the chair, table, desk, bed, bookshelf, sofa, sink, bathtub, toilet, door, and curtain. We take the 9843 models as the referring models in RSL.

ScanNet ScanNetV2 contains 1603 scans, where 1201 scans for training, 312 scans for validation and 100 scans for testing. The 100 testing scans are used for the benchmark, and their labels are unaccessible. We take the 1201 scans to mix up with the referring models for training, and the rest of the 312 scans are used to evaluate the performance.

Table 1: Evaluation on the ScanNet. MinkUNnet is the baseline method. DA, Mix and FA donate data alignment, point mixing up and feature alignment by the prototypes, respectively. GT indicates training with ground truth.

Method	AmAP	chair	bookshelf	sofa	table
MinkUNet	17.81	13.91	31.69	10.44	15.20
MinkUNet+DA$_{noCropping}$	21.28	18.42	39.68	14.41	12.60
MinkUNet+DA	22.91	16.24	41.92	20.32	13.18
MinkUNet+Mix	32.91	46.24	32.32	22.51	30.56
MinkUNet+DA+Mix$_{noNega}$	41.01	54.71	32.71	47.55	25.10
MinkUNet+DA+Mix$_{negaSc}$	29.65	31.34	32.60	26.74	27.93
MinkUNet+DA+Mix$_{negaMo}$	23.14	16.73	43.73	16.38	15.74
MinkUNet+DA+Mix	49.45	66.63	49.54	45.14	36.48
MinkUNet+DA+Mix+FA$_{K48}$	51.19	58.78	**55.79**	49.32	40.90
MinkUNet+DA+Mix+FA$_{K128}$	54.82	**68.54**	50.13	52.41	48.20
MinkUNet+DA+Mix+FA$_{K192}$	51.22	65.47	41.82	51.76	45.82
MinkUNet+DA+Mix+FA$_{T1}$	**55.32**	64.27	51.59	**58.28**	**47.15**
MinkUNet+DA+Mix+FA$_{T8}$	53.85	64.49	53.24	55.33	42.33
MinkUNet+DA+Mix+FA$_{cos}$	53.10	60.41	47.34	62.40	42.24
MinkUNet+GT	81.55	91.03	75.55	85.05	74.56
MinkUNet+DA+Mix+FA$_{K128}$+GT	83.02	92.20	78.89	87.67	73.32

4.2 Evaluation Metric

The goal is to detect the objects (point clouds) that belong to the same class with referring models. Therefore, we calculate the class-specific point-wise possibility on the scan and adopt the mean Average Precision (mAP) to measure the performance for each class. AmAP is the average mAP of all classes.

Table 2: The performance when fine-tuning on the labelled data. We omit the % to show the performance. † means that the network is pre-trained on the referring models by our method. @x% indicates x% of labelled data are used for fine-tuning. The number in () donates the improved accuracy compared with purely supervised training.

Method	mIoU	chair	bookshelf	sofa	table	others
MinkUNet@5%	50.24	77.9	65.7	57.4	57.7	44.6
MinkUNet†@5%	56.12(5.88)	86.1(8.2)	71.0(5.3)	79.3(**21.9**)	64.3(6.6)	48.1(3.5)
MinkUNet@10%	54.86	82.7	66.5	75.1	60.9	50.7
MinkUNet†@10%	59.03(4.17)	86.5(3.8)	72.8(6.3)	79.8(4.7)	66.3(5.4)	54.7(4)
MinkUNet@50%	59.76	86.0	66.5	75.1	66.2	56.3
MinkUNet†@50%	62.00(2.24)	87.3(1.3)	68.9(2.4)	76.7(1.6)	69.1(2.9)	58.6(2.3)
MinkUNet@100%	63.05	89.2	73.8	80.0	69.4	59.2
MinkUNet†@100%	64.93(1.88)	91.1(1.9)	75.3(1.5)	81.5(1.5)	71.7(1.7)	61.1(1.9)

4.3 Implementation Details

We adopt MinkowskiNet14 (Choy et al. (2019)) as the backbone to extract the point-wise feature. Thus, the feature dimension D set to be is 96. The key $\theta(\cdot)$ and the query $\varphi(\cdot)$ function are linear transformation and output 16-dimensional vectors. The voxel size of all experiments is set to be 5 cm for efficient training. Our method is built on the Pytorch platform, optimized by Adam with the default configuration. The batch size for the ScanNet and ModelNet are 4 and 20, respectively, indicating that one scan is mixed up with five referring models. Since there is no colour in the referring models, we set the feature in the Scannet dataset to be a fixed tensor (1), identical to that in the ModelNet. Training 200 epochs cost 20 hours on a GTX 2080 TI GPU. During training, we randomly rotate the models and scans along the z-axis, randomly scales the model and scan with scaling factor 0.9-1.1, and randomly displace the model's location within the scan. If the model's points overlap with the scan's points, we randomly filter the overlapped points or maintain them. We take the chair, bookshelf, sofa and table as foreground classes to evaluate the performance and utilize the remaining classes as background for contrastive learning.

4.4 Results and discussion

In this section, we report the performance by only training the referring models and the performance improvements by fine-tuning the network on the labelled data. Besides, we discuss the potential directions for future works.

Baseline To build the baseline method (MinkUNet in Table 1) for comparison, we first manually align the referring models to the same scale with the objects in the scene, then extract the point feature for individual models and classify the points based on the model tags. During the inferring phase, we directly apply the trained network on the scan for point-wise classification. We treat class separately and use the mAP as the evaluation metric.

Self-supervised Learning with Referring Models The training process of our full method (MinkUNet+DA+Mix+FA) is shown in Figure 1. By contrastive learning the referring models without any annotated scene scans, our method achieves 55.32% AmAP to identify four types of objects on the ScanNet validation dataset, including the chair, bookshelf, sofa and table. Compared with the baseline method, the improvements for the individual class are 51%, 20%, 48% and 32%, respectively. We also show the upper bound performance by training on the annotated scans (MinkUNet+GT), which is 81.55% AmAP, indicating that there is still much room for improvement. When training on both referring models and ground truth (MinkUNet+DA+Mix+FA$_{K128}$+GT), the performance is higher than that only training on ground truth. The qualitative evaluation is shown in Figure 2, indicating the network is well adaptive to the scenes, even only training on the referring models. More cases could be found in supplementary materials.

Fine-tuning with Labeled Scans Our method is beneficial for the downstream task. As shown in table 2, we first pre-train the network by the referring models and then fine-tune the network with

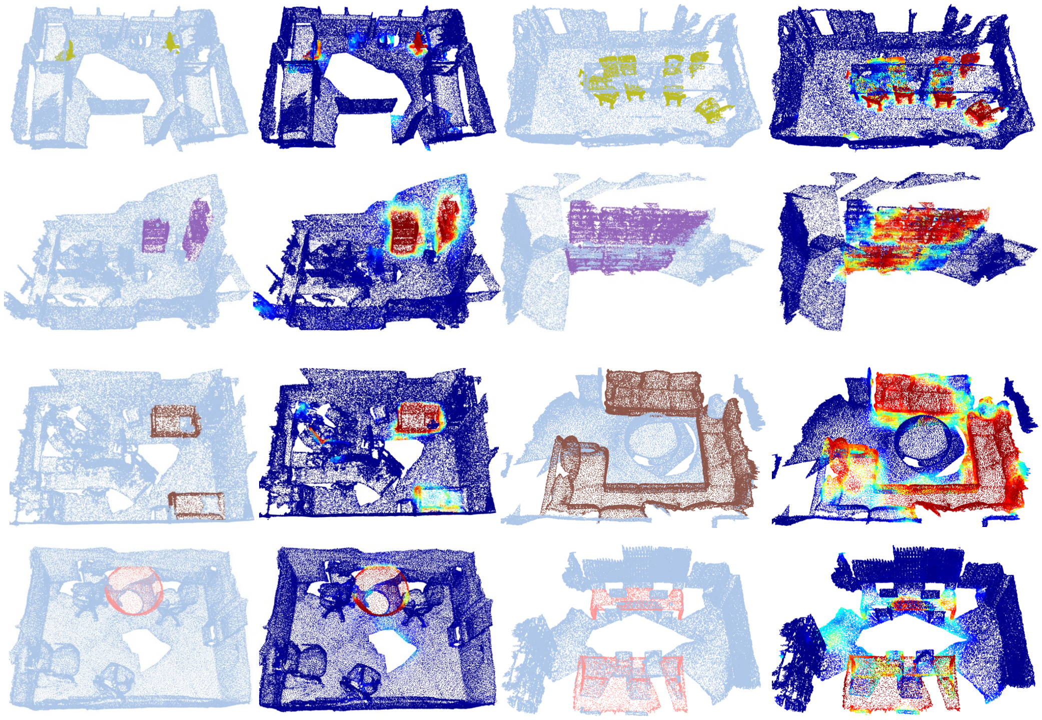

Figure 2: Visualization of inferring on an unseen scene by our method. We show the pairs of ground truth (left) and the inferring results (right). From top to bottom are chair, bookshelf, sofa and table, respectively.

different proportions of labelled scans for scene segmentation, where @5% indicates 5% of labels are used for fine-tuning and † means that the network is pre-trained with the referring models. We adopt mIoU as the evaluation metric. As shown in Table 2, compared and purely supervised counterparts (MinkUNet@5%, MinkUNet@10%, MinkUNet@50%, MinkUNet@100%), the significant improvement could be observed when the network is first pre-trained on the referring models. It is because that the network learns meaningful features from the referring models. Another reason is that the referring models alleviate the long-tail distribution issue. For example, refer to the sofa from the settings MinkUNet@5% and MinkUNet†@5%, the IoU is improved by 21.9%.

Discussions In the current implementation of the Referring Self-supervised Learning, only synthetic 3D models are used for training networks. Since 3D models only provide 3D geometry information, other modalities of data may provide supplementary information for more accurate perception in the real world, such as the colour information by images and semantic information by word embedding. We left it in future work.

4.5 Ablation study

We conduct experiments on ScanNet to verify the effectiveness of different components in our method, including data alignment (DA), point mixing up (Mix), and feature alignment by the prototypes (FA). In the following, we present the configuration details and give more insights into what factors affect the performance.

Effect of Data Alignment MinkUNet+DA indicates that DA is applied on the baseline, including random scaling, rotation, and cropping. These operations aim to cover the diversities of the objects in a real scan. $\text{DA}_{noCropping}$ is the data alignments without random cropping. As shown in Table 1, the performance greatly improved with the use of DA (MinkUNet (17.81 AmAP) VS MinkUNet+DA (22.91 AmAP) and MinkUNet+Mix(32.91 AmAP) VS MinkUNet+DA+Mix (49.45 AmAP)). Besides, we find the random cropping is beneficial for performance improvement because the objects in the real scan are often partially observed. To summary, suitable data alignment that makes synthetic models more realistic is critical for the network to learn valid features that can be adaptive to the real scenes.

Effect of Mixing Up MinkUNet+Mix donates the point mix up is used in the baseline. Observing from (MinkUNet, MinkUNet+Mix) and (MinkUNet+DA, MinkUNet+DA+Mix), the AmAP improved by 15% and 27%, respectively. By mixing up the models and the scene scans, the model feature is augmented by the surrounding objects, closer to the real objects in an unseen scene. Thus, the performance is improved accordingly.

We dig into the details of the mixing up by exploring the following configurations. Firstly, we take out the negative samples in the mixup operation Mix$_{noNega}$, so the network is only trained on foreground classes of referring models. By comparing MinkUNet+DA+Mix and MinkUNet+DA+Mix$_{noNega}$, the performance is decreased by 8%, indicating that contrasting negative samples is also critical to distinguish the foreground classes. It is because that there are similar characters that exist in different classes of models, for example, the right angle in bed and sofa, which fuse the network. Therefore, contrastive learning negative samples force the point feature to encode the contextual semantic feature, reducing the false positive rate.

Secondly, to explore how to conduct negative samples, we try to take the points from the scene as the negative samples for contrastive learning (MinkUNet+DA+Mix$_{negaSc}$). We find that the performance (29.65 AmAP) is significantly worse than MinkUNet+DA+Mix (49.45 AmAP), which uses the referring models as negative samples. It is probably that the network learns the artefacts to distinguish the referring models from the scene. The artefacts are caused by the mixing up operation, such as the overlapped/disjointed point cloud. As a result, the network could not generalize the knowledge to a clear scene without such artefacts. Therefore, by contrastive learning on the positive and negative referring models that both with artefacts, the network learns geometric features for classification.

Lastly, to understanding the role of mixing scene, we only mix up the referring models together and excludes the scan points in the configuration of MinkUNet+DA+Mix$_{negaMo}$. The performance is significantly lower than MinkUNet+DA+Mix (23.14 AmAP VS 49.45 AmAP). It shows that the realistic background is critical for the network to infer an unseen scene.

Effect of Feature Alignment Since feature domain gaps exist between the referring models and the objects in a scan, we use the prototypes to align their features into unified feature space in the configuration MinkUNet+DA+Mix+FA. The experiment shows that the improvement is about 6% for AmAP (MinkUNet+DA+Mix VS MinkUNet+DA+Mix+FA$_K$128). The inversed temperature λ is hyper-parameters for the feature mapping module, indicating the smoothness of the coefficients. We present the results when λ is 1, 4 and 8, respectively. (MinkUNet+DA+Mix+FA$_{T1}$, MinkUNet+DA+Mix+FA$_{K128}$ and MinkUNet+DA+Mix+FA$_{T8}$). Observing the experiments, we find that a more smooth coefficient achieves a better AmAP.

The number of prototypes K is another hyper-parameter. We respectively evaluate them with the configurations MinkUNet+DA+Mix+FA$_{K48}$, MinkUNet+DA+Mix+FA$_{K128}$, and MinkUNet+DA+Mix+FA$_{K192}$. The network achieves the best performance when K is set to be 128. Besides, we show the result when the key $\theta(\cdot)$ and the query function $\varphi(\cdot)$ are identity mapping function in the setting MinkUNet+DA+Mix+FA$_{cos}$.

5 Conclusion

We study a new problem named Referring Self-supervised Learning (RSL) to explore the neural network's cross-scene and cross-domain generalization ability. To solve the issues raised by this problem, we propose a simple yet efficient framework that consists of physical data alignment and convex-hull regularized feature alignment. Like a human behaved in the real world, the neural network recognizes specific objects in a real unseen scene by only learning from the synthetic referring models. Besides, Considering RSL as a pretext of representation learning, the performance is significantly improved in the downstream task.

REFERENCES

Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. *arXiv preprint arXiv:2108.02350*, 2021.

Yunlu Chen, Vincent Tao Hu, Efstratios Gavves, Thomas Mensink, Pascal Mettes, Pengwan Yang, and Cees GM Snoek. Pointmixup: Augmentation for point clouds. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pp. 330–345. Springer, 2020.

Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *NIPS*, 2015.

Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3075–3084, 2019.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017.

Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1705–1714, 2019.

Ben Graham. Sparse 3d convolutional neural networks. *arXiv preprint arXiv:1505.02890*, 2015.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Tong He, Dong Gong, Zhi Tian, and Chunhua Shen. Learning and memorizing representative prototypes for 3d point cloud semantic and instance segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pp. 564–580. Springer, 2020.

Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15587–15597, 2021.

Ruihui Li, Xianzhi Li, Pheng-Ann Heng, and Chi-Wing Fu. Pointaugment: an auto-augmentation framework for point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6378–6387, 2020.

Zhichao Li, Feng Wang, and Naiyan Wang. Lidar r-cnn: An efficient and universal 3d object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7546–7555, 2021.

Hongbin Liu, Jinyuan Jia, and Neil Zhenqiang Gong. Pointguard: Provably robust 3d point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6186–6195, 2021.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2537–2546, 2019.

Fan Lu, Guang Chen, Yinlong Liu, Lijun Zhang, Sanqing Qu, Shu Liu, and Rongqi Gu. Hregnet: A hierarchical network for large-scale outdoor lidar point cloud registration. *arXiv preprint arXiv:2107.11992*, 2021.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 181–196, 2018.

Arthur Ouaknine, Alasdair Newson, Patrick Pérez, Florence Tupin, and Julien Rebut. Multi-view radar semantic segmentation. *arXiv preprint arXiv:2103.16214*, 2021.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017a.

Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017b.

Ralph Tyrell Rockafellar. *Convex analysis*. Princeton university press, 2015.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pp. 213–226. Springer, 2010.

Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2530–2539, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2589–2597, 2018.

Zhonghao Wang, Yunchao Wei, Rogerio Feris, Jinjun Xiong, Wen-Mei Hwu, Thomas S Huang, and Honghui Shi. Alleviating semantic-level shift: A semi-supervised domain adaptation method for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 936–937, 2020.

Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9621–9630, 2019.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *European Conference on Computer Vision*, pp. 574–591. Springer, 2020.

Zhangsihao Yang, Or Litany, Tolga Birdal, Srinath Sridhar, and Leonidas Guibas. Continuous geodesic convolutions for learning on 3d shapes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 134–144, 2021.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *arXiv preprint arXiv:1411.1792*, 2014.

Zhongjie Yu, Lin Chen, Zhongwei Cheng, and Jiebo Luo. Transmatch: A transfer-learning scheme for semi-supervised few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12856–12864, 2020.

Cem Yuksel. Sample elimination for generating poisson disk sample sets. In *Computer Graphics Forum*, volume 34, pp. 25–32. Wiley Online Library, 2015.

Yiming Zeng, Yue Qian, Zhiyu Zhu, Junhui Hou, Hui Yuan, and Ying He. Cornet3d: Unsupervised end-to-end learning of dense correspondence for 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6052–6061, 2021.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Jinlai Zhang, Lyujie Chen, Bo Ouyang, Binbin Liu, Jihong Zhu, Yujing Chen, Yanmei Meng, and Danfeng Wu. Pointcutmix: Regularization strategy for point cloud classification. *arXiv preprint arXiv:2101.01461*, 2021.

Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Few-shot 3d point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8873–8882, 2021.

Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Wei Li, Yuexin Ma, Hongsheng Li, Ruigang Yang, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar-based perception. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.