Learning and Memorizing Representative Prototypes for 3D Point Cloud Semantic and Instance Segmentation

Tong He, Dong Gong, Zhi Tian, and Chunhua Shen

The University of Adelaide
firstname.lastname@adelaide.edu.au

Abstract. 3D point cloud semantic and instance segmentation are crucial and fundamental for 3D scene understanding. Due to the complex structure, point sets are distributed off-balance and diversely, appearing as both category and pattern imbalance. It has been proved that deep networks can easily forget the non-dominant cases during training, which influences the model generalization and leads to unsatisfactory performance. Although re-weighting on instances may reduce the influence, it is hard to find a balance between the dominant and the non-dominant cases. To tackle the above issue, we propose a memory-augmented network that learns and memorizes the representative prototypes that encode both geometry and semantic information. The prototypes are shared by diverse 3D points and recorded in a universal memory module. During training, the memory slots are dynamically associated with both dominant and non-dominant cases, alleviating the forgetting issue. In testing, the distorted observations and rare cases can thus be augmented by retrieving the stored prototypes, leading to better generalization. Experiments on the benchmarks, *i.e.*, S3DIS and ScanNetV2, show the superiority of our method on both effectiveness and efficiency, which substantially improves the accuracy not only on the entire dataset but also on non-dominant classes and samples.

Keywords: Point cloud; Instance segmentation; Memory network.

1 Introduction

3D scene understanding is important and fundamental for various applications, such as robotics, autonomous driving, and virtual reality. The core tasks include semantic segmentation, and instance segmentation on 3D point clouds, *i.e.*, assigning semantic labels and instance indication labels for each point, respectively. Comparing to the studies on 2D images [2, 5, 12, 13], semantic and instance segmentation on 3D point clouds lag far behind and have just started recently [14, 15, 33, 34, 38, 39].

Based on the pioneering works of PointNet [24] and PointNet++ [26], directly processing point sets becomes simpler, more memory-efficient and flexible than handling the volumetric grids with 3D convolution [14, 21, 37]. Some following

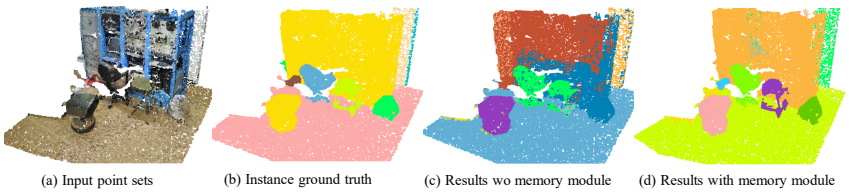

Fig. 1. Comparison of instance segmentation results. The performance of our proposed method shows strong robustness against non-dominant cases.

approaches [33,34,38,39] propose to handle semantic and instance segmentation in an end-to-end network jointly for the fine-grained description of the observation. Specifically, discriminative instance embeddings are learned to measure the instance-level clustering patterns of the points [23,34].

Although existing methods have achieved some impressive results, we still can observe performance bottlenecks on different datasets [1,3], especially on the non-dominant classes with fewer samples (see Fig. 6). It has been proved that deep networks tend to *forget* the rare cases easily while learning on a dataset distributed off balance and diversely [31]. On point cloud data, imbalance issue usually appears as the *category imbalance* and *pattern imbalance*, which is severer than that on 2D images [38]. Defining and measuring the category imbalance is easier, which appears as a significant discrepancy among the proportions of different categories. For example, in an indoor scene (as shown in Fig. 1), the proportions of the points belonging to the background (*e.g.*, wall) are much higher than the objects (*e.g.*, chairs). In S3DIS [1], the total amount of ceiling points is 50 times larger than the chair. The pattern imbalance can be observed on the (non-dominant) rare cases, appearing in both dominant and non-dominant categories, which are usually in the minority of the datasets. It is often caused by complex factors, such as positions, shapes, and relative relationships. For example, chairs are usually placed near a desk, while may occasionally appear with arbitrary positions (e.g., stacking and back-to-back near the cabinet) in an office, as shown in Fig. 1. Conventional methods [38] ignore this issue or simply resort to the focal loss [18], by down weighing the well-learned samples during training. However, it is hard to find a balance between the dominant and non-dominant samples in the dynamic training process.

To address the above issues, we propose to learn and **m**emorize the discriminative and representative **p**rototypes covering all the samples, which is implemented as a memory-augmented network, referred to as **MP**Net. The proposed MPNet includes two branches for predicting point-level semantic labels and obtaining per-point embedding for instance grouping, respectively. As shown in Fig. 2, the two branches access the shared memory via two separate memory readers, associating the two tasks via the shared memory. Given an input, MPNet retrieves the most relevant items in the memory for the extracted per-point features and feeds only retrieved embedding to the following segmentation tasks. In MPNet, the memory is maintained as a compact dictionary shared by diverse points.

Driven by the task-specific training objectives and the proposed geometry-aware regularization, the compact memory is pushed to record the prototypes encoding the geometric and semantic information that is the most representative for all samples. During training, the memory slots are dynamically associated with both the dominant (common) and non-dominant (rare) categories (and cases) seen in mini-batches, alleviating the example forgetting issue [31]. In testing, the distorted observations and rare cases can thus be augmented by retrieving the stored prototypes, leading to better generalization. Additionally, different from previous methods relying on either pairwise relations computation [33] or KNN based feature aggregation [34], the proposed MPNet is free from complex and time-consuming operations, which is more efficient.

The main contributions are summarized as:

- We propose a memory-augmented network (*i.e.*, MPNet) for point cloud segmentation, by learning and memorizing the discriminative and representative prototypes covering all samples. The memory is dynamically associated with both the dominant (common) and non-dominant (rare) categories and cases seen in mini-batch training, alleviating the forgetting issue of the network and leading to better generalization.
- We propose specific regularizations on the memory to learn meaningful and interpretable prototypes in the memory. The proposed memory module is shared by the semantic and instance segmentation task, which naturally associates the two tasks and facilitates the mutual boost.
- The proposed MPNet achieves state-of-the-art performance on large scale datasets, boosting the performance by a large margin not only on the entire dataset but also on the non-dominant classes and samples, with limited consumption on computation and memory.

2 Related Work

Deep Learning for 3D Point Cloud Existing methods for extracting features for 3D point cloud can be roughly categorized into three groups, including voxel-based [21, 37], multi-view based [4, 14, 25, 29] and point-based [16, 24, 26, 30]. [21,37] are the pioneering works to transfer irregular points to regular volumetric grids, aiming to efficiently extract feature representation with 3D convolution. To reduce irrelevant operation on void places and save runtime memory usage, many works are proposed [10, 27]. Multi-view based methods extract features in both 2D and 3D domain. [29] is one of the pioneering multi-view based method, which applies view-pooling over the 2D predictions. 3D-SIS [14], proposed by Hou *et al.* , combines features from 2D and 3D via explicit spatial mapping in an end-to-end trainable network. PointNet [24] is the first deep-learning-based work to operate directly on point sets, which uses shared MLP (multi-layer perceptron) to extract per-point feature. PointNet++ [26] improves the performance by extracting a hierarchical representation. Many following works [16, 17, 30, 32, 36] have been proposed to get a better representation of local context. Due to its simplicity,

we select PointNet++ as our backbone and leave the choices of other backbones for future work.

Instance Segmentation on Point Cloud Deep-learning-based instance segmentation for 3D point cloud is rarely studied until huge application potential has been discovered recently. SGPN [33] is the first deep-learning-based method working on this field. It first splits the whole scene into separate blocks. For every single block, per-point grouping candidates are proposed by predicting a similarity matrix that reflects affinity between each pair of points. A block merging algorithm is conducted for post-processing by considering segmentation results of the overlapped area. However, huge memory is needed for storing the pair-wise matrix, which makes it memory-consuming for post-processing. In order to solve this, Wang *et al.* proposed ASIS [34], which utilized a discriminative loss function [2] to encourage points belonging to the same instance are mapped to a metric space with close distances. Moreover, to make the two tasks take advantage of each other, convolution and KNN search are applied for mutual feature aggregation of the two tasks, making it inefficient and time-consuming.

Memory Networks Memory-based approaches have been discussed for solving various problems. NTM [11] is proposed to improve the generalization ability of the network by introducing an attention-based memory module. Gong *et al.* [9] proposed a memory augmented auto-encoder for anomaly detection, which is detected by represented the input with prototypical elements of the normal data maintained in a memory module. However, the memory model in [9] only includes a single memory pool in autoencoder for unsupervised representation, which may not work for the other tasks. Prototypical Network [28] maintains a category-wise templates for the problem of few-shot classification. Liu [19] proposed an OLTR algorithm to solve the open-ended and long-tail problem by associating a memory feature that can be transferred to both head and tail classes adaptively. These two methods are designed for the task of classification

3 The Proposed Method

3.1 Overview of the Proposed MPNet

We propose a memory-augmented network for joint semantic and instance segmentation in the point cloud data, which learns and memorizes the prototypes of the point sets to alleviate the influence of the imbalanced distribution of the data. As shown in Fig. 2, the proposed memory-augmented network (*i.e.*MPNet) adopts an encoder-decoder architecture, which is free from the specific design of the encoder and decoder. In the proposed MPNet, we use PointNet++ [26] to implement the encoder for per-point feature extraction. Two parallel decoders for instance segmentation and semantic segmentation are built upon the shared encoder. The memory is implemented as a dictionary to record the discriminative and representative prototypes as bases, which are optimized driven by the task-specific objective and the proposed instance-aware geometric regularization.

Given a set of input points $\{\mathbf{p}_i\}_{i=1}^{N}$ with $\mathbf{p}_i \in \mathbb{R}^L$, we can formulate the input of the network as a matrix $\mathbf{P} \in \mathbb{R}^{N \times L}$, where L denotes the input feature

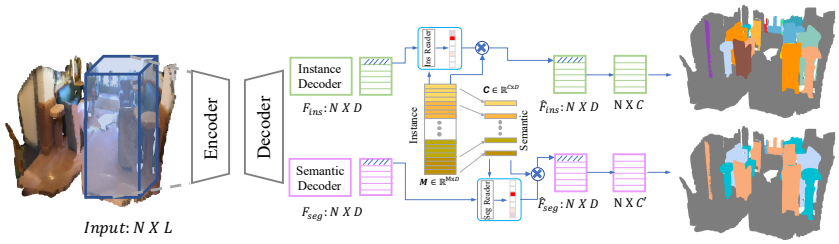

Fig. 2. The framework of our proposed MPNet, which contains two parallel branches with a shared encoder. A memory module is proposed to memorize representative prototypes that are shared by all samples. Both distorted and rare cases can be augmented by retrieving the stored prototypes.

dimension and N denotes the total number of input points. The input features of each point may consist of both geometry and appearance information, *i.e.*, 3D coordinate (x, y, z) and RGB values. The two decoder branches produce features $\mathbf{F}_{\text{seg}} \in \mathbb{R}^{N \times D}$ and $\mathbf{F}_{\text{ins}} \in \mathbb{R}^{N \times D}$, respectively, where D denotes the dimension of the features. Instead of directly using $\mathbf{F}_{\text{seg}}$ and $\mathbf{F}_{\text{ins}}$ to perform semantic and instance segmentation tasks, respectively, MPNet applies them as queries to retrieve the most relevant prototypes in the memory and then obtains features $\widehat{\mathbf{F}}_{\text{seg}}$ and $\widehat{\mathbf{F}}_{\text{ins}}$, which are delivered to the following semantic classifier and instance embedding module. The memory is randomly initialized and optimized during training. The two branches access the memory with specifically designed reading heads.

3.2 Memory Representation for Prototypes

The *prototype memory* is designed as a matrix $\mathbf{M} \in \mathbb{R}^{M \times D}$, where M is a hyper-parameter that defines the number of memory slots and D is the feature dimension that is identical with the outputs from the two branches. The M memory slots are used to restore the prototypes shared by all instances across all categories. To easily represent the semantic characteristics, we define a *semantic memory* $\mathbf{C} \in \mathbb{R}^{C \times D}$, where C denotes the number of categories in semantic segmentation task and each row of $\mathbf{C}$ represents the summary of a class. $\mathbf{C}$ can be seen as the semantic summary of $\mathbf{M}$ and are generated from $\mathbf{M}$. We equally associate the M memory slots in $\mathbf{M}$ with C categories and thus define $M = M_c \times C$, where M_c is the number of per-category prototypes. As shown in Fig. 2, the i-th row in $\mathbf{C}$ is defined as the average of the i-th subsegment (*i.e.*, rows from $(i-1) \times M_c + 1$ to $i \times M_c$) in $\mathbf{M}$:

$$\mathbf{c}_i = \frac{1}{M_c} \sum\nolimits_{j=(i-1)\times M_c+1}^{i \times M_c} \mathbf{m}_j, \tag{1}$$

where $\mathbf{m}_j$ denotes the j-th row vector of $\mathbf{M}$.

Given the query features $\mathbf{F}_{\text{ins}}$ and $\mathbf{F}_{\text{seg}}$, the instance grouping branch directly accesses the prototypes memory $\mathbf{M}$ and the semantic labeling branch accesses

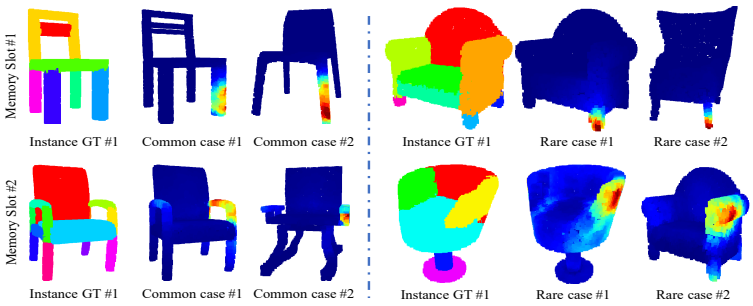

Fig. 3. Visualization of the memory slots in $\mathbf{M}$. We visualize what the memory has learned with the instance segmentation on the PartNet dataset [22], in which the parts (*e.g.*, the chair legs) of the object are treated as instances. For a specific memory slot (*i.e.*, slot #1 and #2 in the figure), we visualize the addressing weights of the points from common and rare cases in pseudo color. The correlation between a specific memory slot and the "visual concepts" (*e.g.*, the components type and relative position) of the most related points are consistent across diverse examples, including common and rare cases, which implies the memory captures meaningful and interpretable semantic and geometric prototypes. Quantitative results are left in the supplementary material.

the semantic summary $\mathbf{C}$, with two specifically designed readers. $\mathbf{M}$ can be seen as a dictionary to restore the representative bases shared by all instances, as the instances cross different categories can share some common basic components and characteristics. Because the semantic memory $\mathbf{C}$ is a re-parameterization of $\mathbf{M}$, the two tasks are naturally associated together, without computation-consuming operations as [34].

3.3 Memory-augmented Instance Embedding

Reading memory for instance embedding Given $\mathbf{F}_{\text{ins}}$, the proposed MPNet reads the most relevant items from $\mathbf{M}$ to obtain instance embedding for instance grouping. For each per-point feature $\mathbf{f}_{\text{ins},i}$ (*i.e.*, the i-th row of $\mathbf{F}_{\text{ins}}$), we calculate the memory addressing weights $\mathbf{w}_i \in \mathbb{R}^M$ according to the similarity between $\mathbf{f}_{\text{ins},i}$ and the prototypes stored in memory $\mathbf{M}$:

$$w_{ij} = \frac{\exp(d(\mathbf{f}_{\text{ins},i}, \mathbf{m}_j))}{\sum_{j=1}^{M} \exp(d(\mathbf{f}_{\text{ins},i}, \mathbf{m}_j))}, \tag{2}$$

where w_{ij} denotes the j-th element of $\mathbf{w}_i$, $\mathbf{m}_j$ is the j-th row in $\mathbf{M}$, and $d(\cdot,\cdot)$ denotes the similarity measurement function. We use cosine similarity as $d(\cdot,\cdot)$. $\mathbf{w}_i$ can also been seen as a soft-attention weight, indicating the relevance of each memory item to the query $\mathbf{f}_{\text{ins},i}$. With $\mathbf{w}_i$, we calibrate $\mathbf{f}_{\text{ins},i}$ with the memory $\mathbf{M}$ and obtain the augmented feature as $\widehat{\mathbf{f}}_{\text{ins},i} = \sum_{j=1}^{M} w_{ij}\mathbf{m}_j$.

Instance-aware geometric regularization Different from previous memory-based representation methods, which are designed for either classification [19] or

unsupervised tasks [9], we propose an instance-aware geometric regularization loss tailored for instance grouping in point cloud, in the hope that the prototypes in the memory module can encode informative geometric information. To achieve this, we force the memory-augmented features from the same instance to have identical geometric predictions.

We first introduce an instance centroids estimator $G(\cdot)$ that will be trained to predict the instance centroids based on the augmented features as $G(\widehat{\mathbf{f}}_{\text{ins},n})$ and try to enforce the predicted centroids to be grouped around the corresponding geometric centers of the instances. The instance-aware regularization loss R_{ins} is defined as:

$$R_{\text{ins}} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{N_k}\sum_{n=1}^{N_k}\|G(\widehat{\mathbf{f}}_{\text{ins},n}) - GT_k\|^2, \tag{3}$$

where K is the instance number, N_k is the number of the points of k-th instance, and GT_k denotes the ground truth centroid of the k-th instance. $\widehat{\mathbf{f}}_{\text{ins},n}$ denotes the augmented feature of a point belonging to the k-th instance. $G(\cdot)$ is implemented as an MLP and can be trained in an end-to-end manner.

What are learnt and stored in memory To have a clear understanding of the learned memory prototypes, we select the category of 'Chair' in PartNet [22] for training and visualization due to its largest number of training samples, as shown in Fig. 3. Quantitative results are presented in the supplementary materials. For each memory item, the points that are addressing it have consistent semantic meaning, implying the capability of the memory module to capture the discriminative and unified representation (for example, position sensitive information) for both common and rare cases.

3.4 Memory-augmented Semantic Labeling

Reading memory for semantic segmentation Similar to the instance grouping branch, the semantic branch reads the category prototypes from semantic memory $\mathbf{C}$ for classification. For each $\mathbf{f}_{\text{seg},i}$ from $\mathbf{F}_{\text{seg}}$, we obtain the soft memory addressing weights $\boldsymbol{\gamma}_i \in \mathbb{R}^C$ by calculating the similarity between $\mathbf{f}_{\text{seg},i}$ and each $\mathbf{c}_j$ (*i.e.*, each row of $\mathbf{C}$), similar to Eq. (2). Then we can obtain $\widehat{\mathbf{F}}_{\text{seg}}$ through $\widehat{\mathbf{f}}_{\text{seg},i} = \boldsymbol{\gamma}_i^{\mathsf{T}}\mathbf{C} = \sum_{j=1}^{C}\gamma_{ij}\mathbf{c}_j$, where γ_{ij} denotes the j-th item in $\boldsymbol{\gamma}_i$.

Semantic memory regularization We apply an additional regularization term on the semantic memory to enforce the centroids of different categories (*i.e.*, the semantic summarization $\mathbf{c}_i$'s) to be separately distributed. Specifically, R_{seg} is used to encourage the augmented feature close to its corresponding category summary in the memory and far away from others. Given $\widehat{\mathbf{f}}_{\text{seg},i}$ and its class label y_i, the regularization term R_{seg} is calculated as:

$$R_{\text{seg}} = \max(0, \sum_{j=y_i}\|\widehat{\mathbf{f}}_{\text{seg},i} - \mathbf{c}_j\| - \sum_{j\neq y_i}\|\widehat{\mathbf{f}}_{\text{seg},i} - \mathbf{c}_j\| + m), \tag{4}$$

where m is the margin, which is set as 5 in our implementation. Each $\mathbf{c}_j$ performs like an anchor point and pulls the features with identical semantic labels close to it and pushes the features with different semantic labels away from it.

3.5 Loss Functions

Classification loss We use the traditional cross-entropy loss L_{ce} for the semantic segmentation task.

Instance discriminative loss Given the per-point memory augmented features $\{\widehat{\mathbf{f}}_{\text{ins},i}\}_{i=1}^{N}$, point-level embeddings $\{\mathbf{g}_{\text{ins},i} \in \mathbb{R}^{c'}\}_{i=1}^{N}$ are generated by a simple MLP layer, where c' is the dimension of the embedding space. Similar to [2,34], we set $c' = 5$ and use the instance discriminative loss for instance grouping. Embeddings from the same instance shoule be grouped together. A soft margin σ_v is introduced to allow these embeddings distributing on a local manifold rather than having to converge to a single point. Moreover, instance embedding centers are no longer repulsed if their distances are larger than $2\sigma_d$. The instance discriminative loss is formulated as:

$$L_{\text{dis}} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{N_k}\sum_{n=1}^{N_k}\left[\|\mathbf{g}_{\text{ins},n} - \boldsymbol{\mu}_k\| - \sigma_v\right]_+^2 + \frac{1}{K(K-1)}\sum_{\substack{i=1\\ i\neq j}}^{K}\sum_{j=1}^{K}\left[2\sigma_d - \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|\right]_+^2, \tag{5}$$

where K is the total instance number, and N_K is the point number of the k-th instance. $\boldsymbol{\mu}_k$ is the average embedding of the k-th instance, which is calculated by $\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{n=1}^{N_k}\mathbf{g}_{\text{ins},n}$. σ_v and σ_d in Eq. (5) are the soft margins. During testing, a simple mean shift clustering algorithm is adopted to group the points in the embedding space.

Training objective As all operations are differentiable, memory can be updated through back-propagation in an end-to-end manner. By combining the four losses discussed above, the training objective is formulated as:

$$L = L_{\text{ce}} + L_{\text{dis}} + R_{\text{seg}} + \lambda R_{\text{ins}}, \tag{6}$$

where λ is the loss weight for R_{ins}. Moreover, as $\mathbf{C}$ is a re-parameterization of $\mathbf{M}$, the supervisions jointly update $\mathbf{M}$ and then influence the two tasks in turn. The two tasks are thus naturally associated together, free from the complex and time-consuming operation, as introduced in [34].

4 Experiments

To validate the effectiveness of our proposed method, both qualitative and quantitative experiments are conducted on two public datasets: Stanford 3D Indoor Semantic Dataset (S3DIS) [1] and ScanNetV2 [3]. The S3DIS dataset [1] is collected in 6 large-scale indoor areas, including 13 classes. ScanNetV2 consists of 1613 indoor scans from 40 categories. The dataset is split into 1201, 312, and 100 for training, validating, and testing, respectively.

4.1 Evaluation

Following [34] on S3DIS dataset, the results on Area-5 and 6-fold cross-validation are reported in our experiments. For semantic segmentation, we present 1) the

Table 1. Ablation study on the S3DIS dataset with vanilla Pointnet++ as backbone. **FL** refers to the focal loss. **InsMem** indicates that the memory is updated by the instance information. **SegMem** means the memory is updated by the supervision from semantic segmentation. **Regul** refers to the regularizations used in learning the prototypes memory. Both instance and semantic segmentation results are provided.

Method	FL	InsMem	SegMem	Regul	mPre	mRec
Baseline					52.3	41.4
	✓				55.2	43.0
		✓			58.9	47.0
		✓	✓		60.2	47.2
Ours		✓	✓	✓	62.5	49.0

overall accuracy (oAcc), which measures the point-level accuracy, 2) the mean class accuracy (mAcc), which calculates the average category-level accuracy, and 3) the instance-wise mean intersection-over-union (mIoU). For instance segmentation, four evaluation metrics are calculated, namely, $mConv$, $mWConv$, $mPrec$, and $mRec$. $mConv$ is defined as the mean instance-wise matching IoU score between the ground truth and the prediction. Instead of treating every instance equally, $mWConv$ is calculated by weighting the size of each instance object. Moreover, traditional $mPrec$ and $mRec$ represent mean precision and recall with the IoU threshold of 0.5, respectively.

4.2 Implementation Details

For the datasets of S3DIS and ScanNetV2, each room is divided into $1m \times 1m$ blocks with a stride of $0.5m$. 4096 points are randomly sampled from each block during the training process. Without special notation, all experiments are conducted using vanilla PointNet++ [26] as backbone (without introducing any multi-scale grouping operation). We utilize the same training setting as ASIS [34]. The whole network is trained in an end-to-end manner for 100 epochs in total. During the inference time, blocks within each room are merged by utilizing the semantic and instance results of the overlapped region. Detailed settings of the algorithm are identical with [33].

4.3 Ablation Study

In this section, we study the influence of each integration of the aforementioned components. All the results are tested on S3DIS Area-5 for a fair comparison. We first build a strong baseline which is equivalent to the vanilla ASIS [34]. Building upon the strong baseline, our MPNet surpasses it by a large margin via memorizing representative prototypes. In the following, we provide detailed analyses on different aspects.

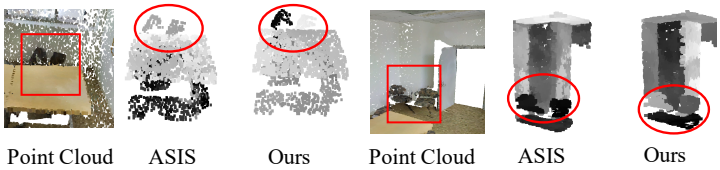

Fig. 4. Barnes-Hut t-SNE [20] visualization of the instance embedding on S3DIS Area-5 set (Best viewed in color and zoomed in).

Memory M and C. The representative and consistent prototypes are maintained in the prototypes memory **M**, which is universal to represent the shared concepts of all instances. Besides, the semantic memory **C** is served as a semantic summary to represent the category characteristics efficiently. As shown in Table 1, using instance memory **M** alone can boost $mPre$ from 52.3% to 58.9% and $mRec$ from 41.4% to 47.0%. On the other hand, using the semantic memory **C** can bring another 1.3% improvements with the metrics of $mPrec$.

Visualizing the Effects of Memory on Instance Embedding In Fig. 4, we directly visualize the instance embedding $\mathbf{g}_{\text{ins},i}$ to show the positive effects of the memory, which covers both the common and rare scenes, *i.e.*office and lobby. The embeddings are projected to 1-D via Barnes-Hut t-SNE [20] for visualization. In both situations, with the help of the memory module, our MPNet generates more discriminative embedding features than the previous state-of-the-art method ASIS [34], which are critical for separating different instances.

Memory Size. We study the influence of the memory size, *i.e.*, the hyper-parameter M or N_c equivalently, to the final performance. The results show that the performance increases as the N_c grows, and becomes stable after when N_c is greater than 200. In all our experiments, N_c is set to 150.

Regularization Loss. To effectively learn representative and discriminative prototypes, regularization losses are proposed in Eq. (3) and Eq. (4), which directly work on the memory-augmented features for instance segmentation and semantic segmentation, respectively. Both of them can be beneficial for both semantic and instance segmentation due to the mutual influence on the memory. As shown in Table 1, the two regularization terms boost the $mPre$ and $mRec$ for about 2.3% and 1.8%, respectively.

Comparing with Focal Loss [18]. The discrepancies among different categories are significant in the 3D point cloud. Focal loss [18] has been widely used in different kinds of vision tasks due to the imbalanced distribution of the training data. It addresses the problem by down-weighting the well-classified samples. However, it only alleviates the category imbalance to some extent and fails to solve the diversely distributed patterns and cases. As shown in Table 1, focal loss can only improve the $mPre$ by 2.9%. Compared with the Focal Loss, our method surpasses the baseline model by a large margin, due to the memorized prototypical patterns and improves $mPre$ and $mRec$ by 8.6% and 5.9%, respectively.

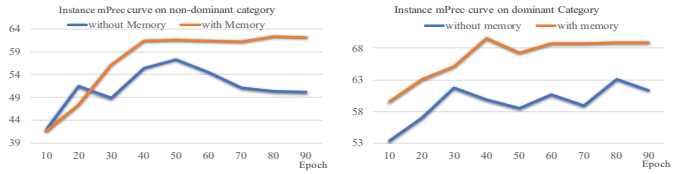

Fig. 5. The training curve on both dominate ("wall") and non-dominant categories ("sofa"). The forgetting issue can be alleviated when associated with our proposed representative memory slots.

Table 2. Instance Segmentation results on S3DIS dataset. Both Area-5 and 6-fold results are reported. All our results are achieved based on a vanilla PointNet++ backbone (without multi-scale grouping) for fair comparison.

Method	Year	mCov	mWCov	mPrec	mRec
Test on Area 5					
SGPN [33]	2018	32.7	35.5	36.0	28.7
ASIS [34]	2019	44.6	47.8	55.3	42.4
3D-BoNet [38]	2019	-	-	57.5	40.2
JSNet [40]	2019	48.7	51.5	62.1	46.9
Ours	-	**50.1**	**53.2**	**62.5**	**49.0**
Test on 6-fold					
SGPN [33]	2018	37.9	40.8	31.2	38.2
MT-PNet [23]	2019	-	-	24.9	-
MV-CRF [23]	2019	-	-	36.3	-
ASIS [34]	2019	51.2	55.1	63.6	47.5
3D-BoNet [38]	2019	-	-	65.6	47.6
PartNet [22]	2019	-	-	56.4	43.4
JSNet [40]	2019	54.1	58.0	66.9	**53.9**
Ours	-	**55.8**	**59.7**	**68.4**	53.7

4.4 Analysis on the Non-dominant Categories and Rare Cases

We study the instance segmentation performance gain brought by the proposed memory network specifically on the non-dominant categories and rare cases.

Analysis on Non-dominant (Rare) Categories. We compare the performance of our proposed MPNet with ASIS [34] on non-dominant classes. We first sort the 13 categories on S3DIS according to their proportions in the training set and split the dataset into three levels: dominant classes (the first 4 classes), mid-dominant classes (the mid 5 classes), and non-dominant classes (the last 4 classes). The amount proportions of the three levels are 79.17%, 16.95%, and 3.88%, respectively. As shown in Fig. 6, we report the improvements with two metrics $mPrec$ and $mRec$. Our method not only boosts the overall performances

Table 3. Comparison per-class performance of our proposed method with the state-of-the-art methods on the task of semantic segmentation on S3DIS. We use vanilla pointnet++ [26] without multi-scale grouping. Even with a simple backbone, the proposed method surpasses the graph-based method by more than 1% mIOU (reported with 6-fold cross-validation).

	OA	miou	ceil	floor	wall	beam	colu	wind	door	table	chair	sofa	book	boar	clut
[24]	78.5	47.6	88.0	88.7	69.3	42.4	23.1	47.5	51.6	54.1	42.0	9.6	38.2	29.4	35.2
[7]	81.1	49.7	90.3	92.1	67.9	44.7	24.2	52.3	51.2	58.1	47.4	6.9	39.0	30.0	41.9
[26]	-	53.2	90.2	91.7	73.1	42.7	21.2	49.7	42.3	62.7	59.0	19.6	45.8	48.2	45.6
[8]	-	58.3	92.1	90.4	**78.5**	37.8	35.7	51.2	65.4	64.0	61.6	25.6	51.6	49.9	53.7
[35]	84.1	56.1	-	-	-	-	-	-	-	-	-	-	-	-	-
[16]	85.9	60.0	93.1	95.3	78.2	33.9	**37.4**	56.1	**68.2**	64.9	61.0	34.6	51.5	51.1	**54.4**
Ours	**86.8**	**61.3**	**94.0**	94.1	76.6	**53.4**	33.6	54.2	62.7	**70.2**	60.2	**36.6**	53.4	**54.3**	53.5

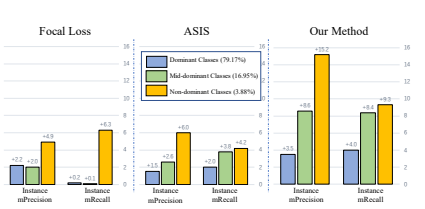

Fig. 6. The comparison of the improvements on both common and uncommon categories. We compare the performance of $mPrec$ and $mRec$ with Focal Loss [18] and ASIS [34].

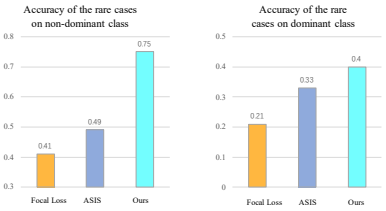

Fig. 7. The instance precision of the rare cases. Both common and uncommon categories are presented. The rare instances are collected as the 20% hardest samples from the baseline model.

but also brings much more significant improvements to the non-dominant classes than focal loss [18] and ASIS [34].

In Fig. 5, we plot the changes of the instance $mPrec$ scores of the model with or without the memory module during training. The results on both the common category ("wall") and uncommon category ("sofa") from S3DIS are shown. With the proposed memory module, our method can alleviate the forgetting issue on the non-dominant samples.

Analysis on Rare Cases. Analyzing with the rare cases is not as easy as on the rare classes since it is not easy to define. We maintain a set of rare cases from "Area-5" in S3DIS [1] by using the performance of the baseline model as the criterion. Specifically, we evaluate the instance-wise IoU score of vanilla ASIS [34] and collect 20% of the instances with the lowest scores as the rare cases for further studies. in Fig. 7, we show the performance of different methods on the rare cases from both a non-dominant class ("sofa") and a dominant class ("wall"). As shown in the figure, the proposed method is more effective to handle

the rare cases. It brings much more improvements than other methods, especially on the rare cases from the non-dominant class, which has more diverse patterns.

4.5 Comparison with the State-of-the-art Methods

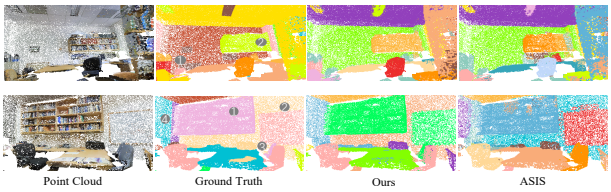

Fig. 8. Qualitative results of our method on S3DIS dataset. From left to right are: input point cloud, instance segmentation ground truth, the results of our method and the results of [34].

Performance on S3DIS. We first compare the instance segmentation performance on both Area-5 and 6-fold. The results are presented in Table 2. Our proposed MPNet achieves promising results and surpasses the previous state-of-the-art approaches substantially by a large margin. The large improvements are mainly beneficial from the strong ability of the proposed memory module. Qualitative results are shown in Fig. 8. In addition to instance segmentation, we also report the results of semantic segmentation and compare them with other methods. The performance is tested on all areas (6-fold), as shown in Table 3. Although based on a simple PointNet++ backbone, we achieve better results than the other methods which are based on graph neural networks [16, 35].
Performance on ScanNetV2. In addition to S3DIS, we conduct experiments on ScanNetV2 [3]. The instance segmentation results are reported in Table 4, which is tested on the validation set. To make a fair comparison, we select the methods that are based on PointNet or PointNet++. Our proposed MPNet outperforms previous methods and dominants in many categories.
Speed Analysis. We compare the inference speed with the other two methods: SGPN [33] and ASIS [34]. The whole evaluation process includes two parts: the network forward and instance grouping. The first part is to get per-point semantic labeling and instance embedding. The second part utilizes a grouping algorithm to find out instance groups. SGPN, which is based on PointNet, predicts a pair-wise affinity matrix to group points into instance clusters, requiring a huge memory buffer. Different from SGPN, ASIS utilizes mean-shift for clustering embeddings to instance groups. Meanwhile, ASIS applies KNN for fusing semantic context from a fixed number of neighboring points. This operation is extremely time-consuming. Compared with the above two approaches, our proposed MPNet is free from complex and time-consuming operations, showing superiority in both effectiveness and efficiency.

Table 4. Instance segmentation results on ScannetV2 benchmark (validation set). The results of mAP@0.25 and mAP@0.5 are reported. All methods except [8] are point-based. (Due to the limited space, Table, Toilet, and Sofa are not presented.)

	mAP @0.25	mAP @0.5	bat.	bed	she.	cab.	cha.	cou.	cur.	des	doo	oth.	pic.	ref.	sho.	sin	sof
[12]	26.1	5.8	33.3	0.2	0.0	5.3	0.2	0.2	2.1	0.0	4.5	2.4	23.8	6.5	0.0	1.4	10.7
[33]	35.1	14.3	20.8	39.0	16.9	6.5	27.5	2.9	6.9	0.0	8.7	4.3	1.4	2.7	0.0	11.2	35.1
[6]	-	24.8	66.7	56.6	7.6	3.5	39.4	2.7	3.5	9.8	9.9	3.0	2.5	9.8	37.5	12.6	60.4
[39]	40.0	23.5	51.3	52.3	12.5	15.2	61.8	0.0	1.5	7.6	29.0	11.7	14.7	25.0	3.7	14.0	34.5
[34]	41.5	24.0	29.9	50.5	0.0	16.7	57.7	0.0	18.4	7.8	14.8	12.9	1.8	12.4	38.0	10.2	36.9
ours	49.3	31.0	69.4	59.8	2.7	23.7	71.1	4.5	8.4	18.3	11.6	17.3	4.8	21.8	57.0	13.4	27.7

Table 5. Inferencing time comparison on the S3DIS Area-5 set. Forward time is the network running time on GPU, whereas postprocessing time is the BlockMerging algorithm introduced in [33]. ASIS is 45% slower than our method in the forward process due to the usage of KNN, which is extremely time-consuming. The reported time is running on a single 1080ti GPU with 4096 input points.

Method	Backbone	Inference Time (ms)			mPre	mRec
		Overall	Forward	Post		
SGPN [33]	PointNet	730	**22**	708	36.0	28.7
ASIS [34]	PointNet2	183	58	**125**	55.3	42.4
Ours	PointNet2	**165**	40	**125**	**62.5**	**49.0**

5 Conclusion

In this paper, we propose a memory-augmented network to handle both category and pattern imbalance in the task of point cloud instance and semantic segmentation. Our method shows superiority in both effectiveness and efficiency.

Acknowledgment

Tong He and Dong Gong contributed equally. Chunhua Shen is the corresponding author.

References

1. Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S.: 3D Semantic Parsing of Large-Scale Indoor Spaces. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2016)
2. Brabandere, B.D., Neven, D., Gool, L.V.: Semantic Instance Segmentation with a Discriminative Loss Function . In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2017)
3. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2017)
4. Dai, A., Nießner, M.: 3DMV: Joint 3D-Multi-View Prediction for 3D Semantic Scene Segmentation. In: Proc. Eur. Conf. Comp. Vis. (2018)
5. Dai, J., He, K., Sun, J.: Instance-aware semantic Segmentation via Multi-task Network Cascades . In: Proc. Eur. Conf. Comp. Vis. (2016)
6. Elich, C., Engelmann, F., Kontogianni, T., Leibe, B.: 3D-BEVIS: Bird's-Eye-View Instance Segmentation . arXiv preprint arXiv:1904.02199 (2019)
7. Engelmann, F., Kontogianni, T., Hermans, A., Leibe, B.: Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds . In: Proc. IEEE Int. Conf. Comp. Vis. Workshops (2017)
8. Engelmann, F., Kontogianni, T., Schult, J., Leibe, B.: Know What Your Neighbors Do: 3D Semantic Segmentation of Point Clouds . arXiv:1810.01151 (2018)
9. Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., van den Hengel, A.: Memorizing Normality to Detect Anomaly: Memory-augmented Deep Autoencoder for Unsupervised Anomaly Detection. In: Proc. IEEE Int. Conf. Comp. Vis. (2019)
10. Graham, B., Engelcke, M., van der Maaten, L.: 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2018)
11. Graves, A., Wayne, G., Danihelk, I.: Neural Turing Machines . arXiv preprint arXiv:1410.5401 (2014)
12. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN . In: Proc. IEEE Int. Conf. Comp. Vis. (2017)
13. He, T., Shen, C., Tian, Z., Gong, D., Sun, C., Yan, Y.: Knowledge adaptation for efficient semantic segmentation. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2019)
14. Hou, J., Dai, A., Nießner, M.: 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2019)
15. Lahoud, J., Ghanem, B., Pollefeys, M., Oswald, M.R.: 3D Instance Segmentation via Multi-task Metric Learning. arXiv preprint arXiv:1906.08650 (2019)
16. Li, G., Mller, M., Thabet, A., Ghanem, B.: DeepGCNs: Can GCNs Go as Deep as CNNs? In: Proc. IEEE Int. Conf. Comp. Vis. (2019)
17. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: PointCNN: Convolution On X-Transformed Points. In: Proc. Advances in Neural Inf. Process. Syst. (2018)
18. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal Loss for Dense Object Detection. In: Proc. IEEE Int. Conf. Comp. Vis. (2017)
19. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-Scale Long-Tailed Recognition in an Open World. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2019)

20. van der Maaten, L.: Accelerating t-SNE using Tree-Based Algorithms. J. Mach. Learn. Res. **15**, 3221–3245 (2014), http://jmlr.org/papers/v15/vandermaaten14a.html
21. Maturana, D., Scherer, S.: VoxNet: A 3D Convolutional Neural Network for real-time object recognition. In: Proc. IEEE Int. Conf. Intelligent Robots Syst. (2015)
22. Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H.: PartNet: A Large-scale Benchmark for Fine-grained and Hierarchical Part-level 3D Object Understanding. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2019)
23. Pham, Q.H., Nguyen, D.T., Hua, B.S., Roig, G., Yeung, S.K.: JSIS3D: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2019)
24. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2017)
25. Qi, C.R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L.J.: Volumetric and Multi-View CNNs for Object Classification on 3D Data . In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2016)
26. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In: Proc. Advances in Neural Inf. Process. Syst. (2017)
27. Riegler, G., Ulusoy, A.O., Geiger, A.: OctNet: Learning Deep 3D Representations at High Resolutions . arXiv preprint arXiv:1611.05009 (2016)
28. Snell, J., Swersky, K., Zemel, R.S.: Prototypical Networks for Few-shot Learning . In: Proc. Advances in Neural Inf. Process. Syst. (2017)
29. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view Convolutional Neural Networks for 3D Shape Recognition. In: Proc. IEEE Int. Conf. Comp. Vis. (2015)
30. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: KPConv: Flexible and Deformable Convolution for Point Clouds. In: Proc. IEEE Int. Conf. Comp. Vis. (2019)
31. Toneva, M., Sordoni, A., Combes, R.T.d., Trischler, A., Bengio, Y., Gordon, G.J.: An empirical study of example forgetting during deep neural network learning. arXiv preprint arXiv:1812.05159 (2018)
32. Wang, L., Huang, Y., Hou, Y., Zhang, S., Shan, J.: Graph Attention Convolution for Point Cloud Semantic Segmentation. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2019)
33. Wang, W., Yu, R., Huang, Q., Neumann, U.: SGPN: Similarity Group Proposal Network for 3D Point Cloud Instance Segmentation. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2018)
34. Wang, X., Liu, S., Shen, X., Shen, C., Jia, J.: Associatively Segmenting Instances and Semantics in Point Clouds. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2019)
35. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic Graph CNN for Learning on Point Clouds. ACM Trans. On. Graphic (2019)
36. Wu, W., Qi, Z., Fuxin, L.: PointConv: Deep Convolutional Networks on 3D Point Clouds. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2019)
37. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3D ShapeNets: A Deep Representation for Volumetric Shapes . In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2015)

38. Yang, B., Wang, J., Clark, R., Hu, Q., Wang, S., Markham, A., Trigoni, N.: Learning Object Bounding Boxes for 3D Instance Segmentation on Point Clouds. In: Proc. Advances in Neural Inf. Process. Syst. (2019)
39. Yi, L., Zhao, W., Wang, H., Sung, M., Guibas, L.J.: GSPN: Generative Shape Proposal Network for 3d Instance Segmentation in Point Cloud. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2018)
40. Zhao, L., Tao, W.: JSNet: Joint Instance and Semantic Segmentation of 3D Point Clouds . In: Proc. AAAI Conf. Artificial Intell. (2020)