

Adaptive Margin Based Metric Learning for Unbalanced Point Cloud Classification and Retrieval

Ziqiang Chen

Guilin University of Electronic Technology

Danqi Zheng

Guilin University of Electronic Technology

Chang Liu

Guilin University of Electronic Technology

Lutong Zhang

Guilin University of Electronic Technology

Tao Huang

httz1001@163.com

Xidian University

Research Article

Keywords: 3D object classification and retrieval, Point cloud, Metric learning, Unbalanced dataset

Posted Date: October 15th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-8279902/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Adaptive Margin Based Metric Learning for Unbalanced Point Cloud Classification and Retrieval

Ziqiang Chen¹, Danqi Zheng¹, Chang Liu¹, Lutong Zhang¹, Tao Huang^{2*}

¹Guangxi Key Laboratory of Wireless Broadband Communication and Signal Processing, Guilin University of Electronic Technology, No.1 Jinti Road, Guilin, 541004, Guangxi, China.

²Hangzhou Institute of Technology, Xidian University, No.177 Lvxue Road, Hangzhou, 311200, Zhejiang, China.

*Corresponding author(s). E-mail(s): httz1001@163.com;

Contributing authors: 1420536815@qq.com; 328777140@qq.com; 1160088681@qq.com; 1260509733@qq.com;

Abstract

Loss function plays a crucial role in modern deep learning based 3D object classification and retrieval methods. The widely used softmax loss relies on an implicit hypothesis that all classes have sufficient samples and intraclass feature spaces are separated by a uniform margin. Yet real world large scale 3D object datasets are highly unbalanced making it challenging for softmax loss to simultaneously ensure intraclass compactness and interclass discrepancy. To address this issue we propose an adaptive margin based loss function for point cloud learning from highly unbalanced datasets which to the best of our knowledge is the first work introducing margin based metric learning into point cloud classification and retrieval thus achieving good adaptation to unbalanced 3D datasets. Additionally the loss function incorporates regularization constraints on weight vectors. To enable cross layer feature interaction and enhance network capacity we also design a hierarchical bilinear pooling strategy across layers and a lightweight spatial transform module. Extensive experiments on ModelNet40 and ShapeNetCore55 datasets demonstrate that our method with very low computational cost outperforms state of the art approaches based on multiple 3D object representations.

Keywords: 3D object classification and retrieval, Point cloud, Metric learning, Unbalanced dataset

1 Introduction

In the past decades, 3D vision has played an important role in applications such as autonomous vehicles, human-computer interaction, and interior design [1–3]. Among various 3D vision tasks, object classification and retrieval are among the most widely studied in the literature. Due to the availability of large-scale 3D labeled datasets

such as ModelNet, ShapeNet [4], deep learning for 3D object classification and retrieval has advanced rapidly in recent years. Various methods based on different 3D object representations including multiview, mesh and point cloud [5–10] have been proposed. For multiview-based methods [6, 11, 12], multiple views of 3D objects are captured by cameras from different angles and can

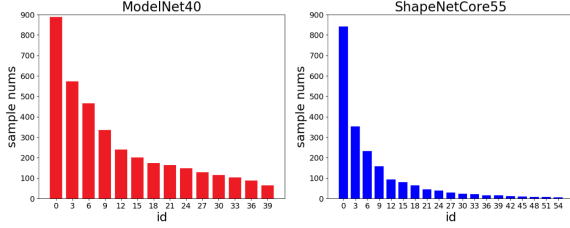


Fig. 1 Motivation behind adaptive marginal loss: highly unbalanced datasets such as *ModelNet40* and *ShapeNetCore55* are characterized by a heavy-tailed distribution of samples per class.

be efficiently processed by off-the-shelf 2D convolutional neural network (CNN). However, one of the remaining key challenges is the requirement of a fixed camera view, which is often impractical in real applications. To address this issue, the point cloud of objects obtained from the 3D sensor such as LiDAR [1] can well preserve the spatial structure of 3D objects and is insensitive to the varying views. To better support the task of 3D shape recognition, multiview and point cloud fusion-based methods [13–15] learn the relation between point cloud and multiview images in a unified learning framework. Despite the remarkable performance of fusion-based methods, they suffer from heavy computational cost and inflexible camera setting problems. Consequently, specially designed neural networks such as PointNet [8] achieve permutation invariance by independently operating each point and extracting features by symmetric functions. Variations such as considering the neighborhood of points [9] and capturing the local geometry [16] have also been developed in the literature.

The motivation behind this work is largely two-fold. On the one hand, most existing works [16–18] train the network in an end-to-end manner by optimizing standard softmax loss function. The implicit hypothesis behind softmax is that all classes have sufficient samples to describe the true distribution of data. Unfortunately, many public datasets in practice are highly unbalanced - i.e., the number of samples per class observes a heavy-tailed distribution as shown in Figure 1. A direct consequence of the unbalanced dataset is that the feature space after learning would be dominated by the few rich classes with a large number of samples. This is because the standard softmax loss

fails to enforce intraclass compactness and inter-class discrepancy simultaneously. It follows that the state-of-art methods have achieved impressive performance on the task of shape classification only (the performance degrades by over 3% on the task of shape retrieval as shown in [13]). Inspired by the success of adaptive margin-based loss function in face recognition (e.g., cosface [19] and arcface [20]), we propose to design a novel strategy for point cloud representation learning, where the margin is adaptively assigned to each class. Unlike the fixed margin methods in [19, 20], we let the margin be inversely proportional to the sample number of each class. To the best of our knowledge, this is the first attempt to introduce metric learning methods into the point cloud learning framework, leading to a good adaptation to the heavy-tailed unbalanced datasets.

On the other hand, there exists some duality between point cloud-based and image-based object recognition. It is natural to leverage ideas from the rich literature of image-based object recognition to the emerging field of point cloud-based one. Inspired by the success of hierarchical bilinear pooling (HBP) in fine-grained visual recognition [21], we propose to capture interlayer part-feature relations for enhancing their representation capability. More specifically, the pooling between the middle and high level layers of the network allows us to capture fine-grained features and promote better feature aggregation. To further exploit the benefit brought by HBP, the regularization constraint for minimizing the inner product between different weight vectors is considered in the loss function. In addition, a new spatial transform module is introduced to transform the input point set into a canonical space with low computational cost, which jointly works with the proposed HBP learning.

2 Related Work

2.1 3D Representation Learning Methods

Research on 3D deep learning has evolved through multiple representation paradigms. Multi-view approaches such as MVCNN [6], VNN [11], and Dist [12] project 3D objects onto 2D planes from different viewpoints and leverage 2D CNNs for feature extraction. These methods require

fixed camera configurations which limits practical deployment. Point cloud based methods directly process raw 3D coordinates. PointNet [8] established permutation invariant processing through symmetric functions, while PointNet++ [9] enhanced local feature learning through hierarchical structures. DGCNN [16], DensePoint [17], and RS-CNN [18] further improved local geometric feature extraction. Recent advances like PointNetXt [22] revisited PointNet++ architecture with improved training strategies, while Point Mamba [10] introduced state space models for efficient point cloud processing. Fusion based approaches including PVRnet [13], FusionNet [14], and PVNet [15] combine multi-view images and point clouds in unified frameworks. Despite strong classification performance, these methods typically optimize using standard softmax loss which fails to enforce intra-class compactness and inter-class discrepancy simultaneously, resulting in suboptimal retrieval performance as evidenced by the over 3% degradation reported in [13].

2.2 Metric Learning and Feature Enhancement

Class imbalance poses significant challenges for 3D shape learning as real-world datasets often follow heavy-tailed distributions. Standard softmax loss tends to bias feature spaces toward majority classes, a limitation particularly evident in methods like DGCNN [16], DensePoint [17], and RS-CNN [18]. In face recognition, margin-based approaches such as CosFace [19] and ArcFace [20] introduced fixed margins to enhance feature discrimination. Their success inspires our adaptive margin strategy for point cloud learning. For feature representation enhancement, hierarchical bilinear pooling [21] has demonstrated success in fine-grained image recognition by modeling pairwise feature interactions across network layers. This technique remains unexplored for point cloud learning despite its potential to enhance the retrieval capabilities of methods like PointNet++ [9] and DGCNN [16]. Our work bridges these gaps by introducing class-adaptive margin learning with hierarchical feature modeling specifically designed for unbalanced 3D datasets, while maintaining computational efficiency compared to multi-view and fusion approaches [6, 13, 15].

3 Proposed method

3.1 Adaptive Margin based loss function

The widely used softmax loss function for 3D object classification and retrieval can be written as [19, 20]

$$L = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{\mathbf{W}_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^n e^{\mathbf{W}_j^T x_i + b_j}}, \quad (1)$$

where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the deep embedding feature of the i -th sample associated with the y_i -th class, $\mathbf{W}_j \in \mathbb{R}^d$ denotes the j -th column of the weight matrix $\mathbf{W} \in \mathbb{R}^{d \times n}$ (d, n denotes the embedding feature dimension and the total class number respectively) from last fully connected layer of deep network, $b_j \in \mathbb{R}^n$ denotes the bias term, and N is the batch size. The softmax loss function learns to classify each training sample to its target class by maximizing the similarity score between its embedding feature x_i and its target proxy $\mathbf{W}_{y_i}$ representing the class y_i , which is also known as the classification loss function [23].

After the l_2 normalization of the embedding feature and each column of weight matrix, the cosine similarity score between x_i and $\mathbf{W}_j$ can be transformed as $\mathbf{W}_j^T x_i = \|\mathbf{W}_j\|_2 \|x_i\|_2 \cos \theta_j = \cos \theta_j$, where θ_j denotes the angle between the feature and the weight vector. For the feature x_i with label y_i , the intraclass score $s_p^{y_i}$ denotes the similarity score between the normalized feature and the weight vector $\mathbf{W}_{y_i}$ corresponding to label y_i and the interclass score s_q^j denotes the similarity score between the normalized feature and the weight vector $\mathbf{W}_j$ corresponding to label j ($j \neq y_i$). Fixing the bias b_j as 0, Eq.(1) can be formulated as

$$L = \frac{1}{N} \sum_{i=1}^N \log(1 + \sum_{j=1}^n \exp(s_q^j - s_p^{y_i})). \quad (2)$$

Following [20], rescaling the cosine similarity and adding a constant margin m are commonly used in the design of modern classification loss

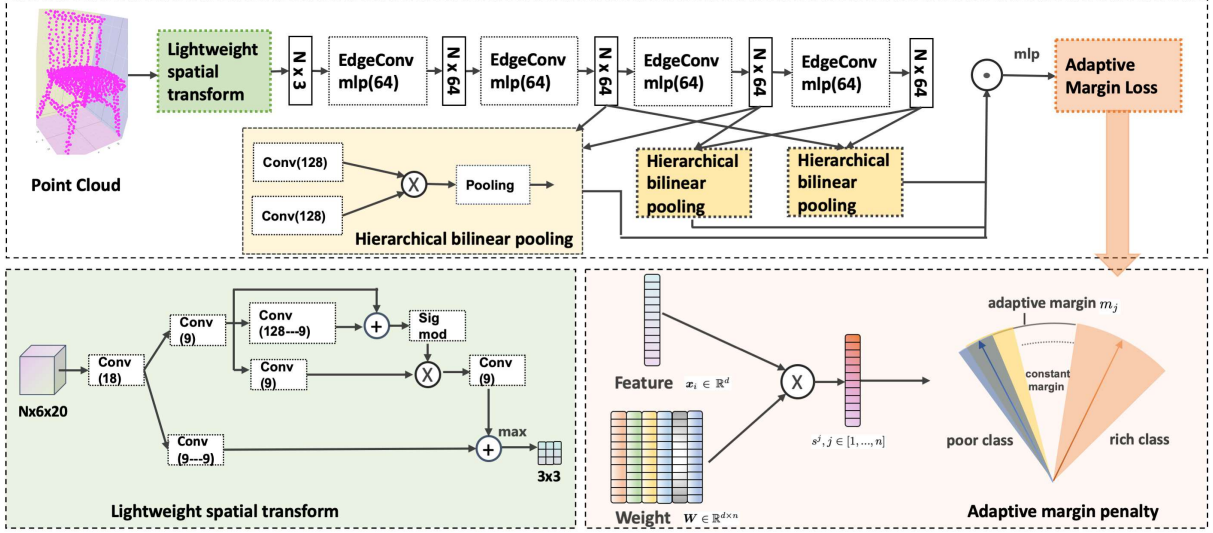


Fig. 2 Proposed network architecture: the backbone network consisting of several *EdgeConv* blocks is based on DGCNN [16]; the lightweight spatial transform (LST) and hierarchical bilinear pooling (HBP) are new modules introduced in this work.

function. Then Eq. (1) can be rewritten into

$$L = \frac{1}{N} \sum_{i=1}^N \log(1 + \sum_{j=1}^n \exp(\gamma(s_q^j - s_p^{y_i} + m))), \quad (3)$$

where γ is the predefined rescaling parameter and the margin m can be interpreted as the penalty between intraclass score $s_p^{y_i}$ and the interclass similarity score s_q^j in the normalized hypersphere. Note that the Eq.(2) of softmax is same as the Eq.(3) when $\gamma=1$ and $m=0$.

In [19, 20], the cosine or angular margin m has to be a constant. However, in practice, 3D object datasets are highly unbalanced as shown in Fig. 1, which has not been carefully considered in previous works on 3D object recognition. For each class with different number of samples, the objective that all classes share the same margin is clearly unreasonable. A large margin should be assigned to those poor classes without sufficient number of samples for interclass separation; while a small margin should be assigned to rich classes for intraclass compactness. Based on the above observations, an adaptive margin that is inversely proportional to the number of samples is proposed

as follows

$$m_j = 1 - \frac{1}{1 + \exp(-\theta_j/\theta_{max})}, \quad (4)$$

where θ_j is the sample number of class j and θ_{max} is the maximum sample number of classes. After the maximum normalization operation, each margin is rescaled by the constant parameter β . By substituting Eq.(4) and β into Eq.(3), we obtain

$$L = \frac{1}{N} \sum_{i=1}^N \log(1 + \sum_{j=1}^n \exp(\gamma(s_q^j - s_p^{y_i} + \beta m_j))). \quad (5)$$

To further separate different classes in the feature space, the orthogonal loss between weight vectors can be added into Eq. (5),

$$L = \frac{1}{N} \sum_{i=1}^N \log(1 + \sum_{j=1}^n \exp(\gamma(s_q^j - s_p^{y_i} + \beta m_j))) + \lambda \log(1 + \frac{1}{N} \sum_{i=1}^N \max_{j \neq y_i} \mathbf{W}_{y_i}^T \mathbf{W}_j + \frac{1}{2}), \quad (6)$$

where λ is the regularization parameter.

3.2 Hierarchical bilinear pooling

For the visual fine-grained recognition task, bilinear pooling learning [21] has been shown to be a powerful tool for feature extraction and aggregation. An important new insight brought by [21] is that interlayer feature interaction and fine-grained feature learning are intrinsically correlated and therefore can benefit each other. Unlike directly concatenating all features, bilinear pooling can integrate multiple cross-layer bilinear features to enhance their representation ability. A hierarchical extension of bilinear pooling allows us to better capture semantic feature relations across different levels, which leads to superior performance for the task of fine-grained visual recognition. Inspired by the success of hierarchical bilinear pooling (HBP), we propose to enable the cross-layer interaction of features by bilinear pooling between output features from the middle and high layers of the network. The detailed architecture is shown in Fig.2, where two input features after convolution projection and Hadamard product operation are mapped in a quadratic expansion way of linear transform to learn a joint representation of the output features.

3.3 Lightweight Spatial transform

The spatial transform module in [16] is proposed to align the input point cloud set to a canonical space by applying an estimated 3×3 matrix and is only designed for the segmentation task. The main purpose of this module is not clearly explained in the original paper, but our explanation is that this module can further enhance the capacity of the network and thus better handle the random rotated point cloud set in ShapeNet dataset [4]. Compared with the origin network of [16] that has 1.83 million (M) parameters, this module dominates nearly half of the whole computational cost, which has 0.81M parameters.

To further reduce the parameter cost and maintain the capacity of network simultaneously, a new lightweight spatial transform (LST) module is proposed as shown in Fig.2. Borrowing the sophisticated design idea of self-calibrated [24], the standard convolution with a larger channel is replaced with the self-calibrated convolution with a smaller channel by splitting the convolutional filter into multiple portions and merging these portions into a more discriminative feature.

Compared with the origin module in [16], our LST module has only 3,579 parameters and the performance is not sacrificed.

The whole network of our method applied in the classification and retrieval of point cloud is shown in Fig.2. The point cloud set is first aligned by the proposed lightweight spatial transform module, and then multiple output embedding features from convolution operations are fused by a hierarchical bilinear pooling module to be a global shape feature. Finally, the adaptive margin loss between the global shape feature and its target is computed with the aim of enforcing intraclass compactness and interclass discrepancy simultaneously. The network is trained in an end-to-end manner.

4 Experimental results

4.1 Implementation Details

All experiments are conducted in pytorch on a computer with two Nvidia GTX2080Ti GPUs and a Intel i7-7800X CPU. The stochastic gradient descent (SGD) optimization algorithm is employed with momentum 0.0001 for training and a mini-batch size of 110. The initial learning rate is 0.1 and the cosine learning rate decay with total epochs 200 and warmup epochs 20 is employed. During the training, a dropout technique with 50% in FC layers is applied. The rescale parameter γ , β and λ is fixed as 9.6, 0.83, and 10, respectively. The 1024 point cloud data of the 3D object is uniformly sampled from the surface of each object model and is normalized to a unit sphere.

4.2 Results on ModelNet40

ModelNet40 dataset [4] consists of 9,843 training models and 2,468 test models across 40 object categories. We evaluate our method on both classification and retrieval benchmarks, comparing against state-of-the-art approaches utilizing different 3D representation modalities. For point cloud-based methods, we compare with PointNet [8], PointNet++ [9], DGCNN [16], RSCNN [18], Densepoint [17], Point2SpatialCapsule [27], PosPool [28], PCSALMIX [29], and the recently proposed Point Mamba [10]. Among multi-view methods, we include MVCNN [6], TCL [25], Ma [5], He [11], ATC [26], VNN [11], MVContrast [7],

Table 1 Classification and retrieval results on the ModelNet40 dataset.

data	Method	Classification (Accuracy)	Retrieval (mAP)
Point Cloud and 12 views	FusionNet [14]	93.2%	89.5%
	PVRNet [13]	93.6%	90.5%
12 views	MVCNN [6]	89.9%	80.2%
	TCL [25]	—	88.0%
	Ma [5]	91.1%	84.3%
	He [11]	—	89.3%
	ATC [26]	—	86.1%
	VNN [11]	—	89.3%
	MVContrast [7]	93.3%	90.3%
	Dist [12]	93.4%	—
Point Cloud	PointNet [8]	89.2%	—
	PointNet++ [9]	90.7%	—
	DGCNN [16]	92.2%	81.6%
	RSCNN [18]	92.7%	86.6%
	Densepoint [17]	92.5%	84.0%
	Point2SpatialCapsule [27]	93.4%	89.4%
	PosPool [28]	93.2%	—
	PCSALMIX [29]	93.6%	—
	Point Mamba [10]	93.4%	—
	Ours	93.7%	92.3%

Table 2 Classification and retrieval results on the ModelNet40 dataset.

Method	Para. (M)	FLOPs	Runtime(ms)	
			GPU	CPU
PVRNet [13]	70	2.52G	9.2	192.1
DGCNN [16]	1.83	1.23G	4.2	164.9
RSCNN [18]	1.91	0.25G*	12.2	—
Densepoint [17]	0.65	0.12G*	6.6	—
Ours	0.43	0.75G	5.6	99.5

and Dist [12]. We also benchmark against fusion-based approaches including FusionNet [14] and PVRNet [13]. Following the evaluation protocol in [13], we use overall accuracy for classification and mean Average Precision (mAP) for retrieval tasks. During training, input data is randomly scaled within [0.66, 1.5] and translated within [-0.2, 0.2]. Following [18], we employ 10-fold voting with random rescaling and average the predictions.

As shown in Table 1, our method achieves superior performance across both tasks among

all point cloud approaches. Notably, we outperform the recent advanced methods including PCSALMIX (+0.1% classification accuracy) and Point Mamba (+0.3% classification accuracy), while achieving a significant margin in retrieval performance. Our method obtains 92.3% mAP, surpassing Point2SpatialCapsule (the previous best point cloud method) by 2.9% and exceeding the strongest fusion-based method (PVRNet) by 1.8%. This performance gap demonstrates our method’s exceptional capability in capturing both global structure and local geometric details for effective shape understanding and retrieval. The consistent superiority across both classification and retrieval tasks validates the effectiveness of our architecture design, particularly in modeling fine-grained geometric features that are crucial for distinguishing similar 3D shapes.

The comparisons about parameters, computational cost, and inference running times on GPU and CPU are summarized in Table 2. In general, the computational cost of the network is dominated by the convolution and linear FC layers. Here we only report the computation cost of

Table 3 Comparison of performance on the ShapeNetCore55 perturbed dataset.

Method	Data	Micro			Macro		
		F-measure	mAP	NDCG	F-measure	mAP	NDCG
Wang [30]	36 views	24.6%	60.0%	77.6%	16.3%	47.8%	69.5%
Li [30]	12 views	53.4%	74.9%	86.5%	18.2%	57.9%	76.7%
MVCNN [6]	80 views	61.2%	73.4%	84.3%	41.6%	66.2%	79.3%
Kd-network [31]	20 views	61.2%	73.4%	81.4%	24.1%	48.4%	79.3%
GIFT [32]	64 views	66.1%	81.1%	88.9%	42.3%	73.0%	84.3%
ATC[26]	8 views	66.5%	84.3%	88.5%	40.4%	77.7%	85.1%
VNN [11]	80 views	71.3%	84.3%	89.7%	50.1%	78.0%	86.8%
BDNet [33]	4 views	46.1%	60.3%	85.7%	18.2%	42.7%	72.9%
VSFormer [34]	20 views	82.0%	79.3%	82.4%	66.7%	64.2%	70.4%
Ours	Point Cloud	72.0%	87.8%	91.3%	47.8%	80.6%	88.1%

convolution and FC layers in RSCNN and Densepoint methods, which has not taken the ball query and grouping operation accelerated by CUDA into account. Therefore, RSCNN and Densepoint methods cannot be directly applied on CPU. From Table 2, it can be seen that our method consumes very few FLOPs with only 0.43M parameters. Fusion-based method PVRNet takes 12 views and 1024 point cloud set as input and thus needs the large computational cost. To sum up, our method has achieved the state-of-art classification and retrieval performance with the least parameters and very few FLOPs.

Table 4 Ablation study of the proposed method.

Method	Classification (Accuracy)	Retrieval (mAP)
Softmax	91.0%	88.3%
AML	92.7%	91.4%
AML+HBP	92.9%	91.5%
AML+HBP+LST(Ours)	93.7%	92.3%
Softmax+HBP+LST	91.6%	89.5%
CosFace+HBP+LST	92.1%	91.5%
ArcFace+HBP+LST	92.4%	91.3%
AML+HBP+LST(Ours)	93.7%	92.3%

4.3 Results on ShapeNetCore55

To demonstrate the robustness of our method on challenging scenarios, we conduct experiments on the perturbed version of the ShapeNetCore55 dataset, where models undergo random rotations. This dataset comprises 35,764 training models,

5,133 validation models, and 10,265 test models spanning 55 object categories, originally released for the SHREC’16 Large-Scale 3D Shape Retrieval competition [30]. We compare our approach against diverse methods including early competition entries (Wang [30], Li [30]), multi-view techniques (MVCNN [6], GIFT [32], Kd-network [31], VNN [11]), and recent advances (ATC [26], BDNet [33], VSFormer [34]). Notably, our work represents the first point cloud-based method evaluated on this challenging benchmark. Table 3 presents comprehensive comparisons across three critical metrics F-measure, mAP, and NDCG, reported in both Micro and Macro averaged formats using the official evaluation protocol [30]. While VSFormer achieves superior F-measure scores (particularly in Micro setting), our method establishes state-of-the-art performance on the more comprehensive metrics of mAP and NDCG across both averaging methods, achieving 87.8% mAP and 91.3% NDCG in Micro evaluation, and 80.6% mAP and 88.1% NDCG in Macro evaluation. This performance pattern reveals an important insight. F-measure primarily evaluates binary relevance at fixed thresholds, whereas mAP and NDCG comprehensively assess the quality of the entire ranked retrieval list with greater emphasis on top-ranked results. The superior performance in these latter metrics demonstrates our method’s exceptional capability in producing highly relevant and properly ordered retrieval results, which is a critical requirement for real-world applications. Furthermore, our method accomplishes this using raw point clouds without requiring expensive multi-view rendering. All competing methods utilize 4 to 80 views, making our approach significantly more

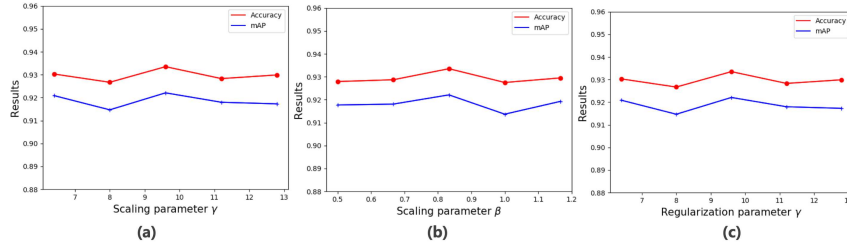


Fig. 3 The classification and retrieval result curves as function of: (a) the re-scaling parameter γ ; (b) the re-scaling parameter β ; and (c) the regularization parameter λ .

efficient and practical for real-time applications. This combination of superior retrieval quality and computational efficiency represents a substantial advancement in robust 3D shape retrieval under perturbation conditions.

4.4 Ablation study

To verify the effectiveness of the three components in the proposed method, we conduct a number of ablations to study the improvements of adaptive margin loss (AML) function, hierarchical bilinear pooling (HBP) and lightweight spatial Transform (LST) respectively. The baseline method (Softmax) is that AML and HBP are replaced with standard softmax and directly connecting, respectively, and LST module is not adopted in Figure 2. From Table 4, it can be seen that the improvement of AML on retrieval performance is most obvious from 88.3% to 91.4%. HBP and ST can also improve the performance with few computational costs when combined with AML. The accuracy and mAP over the baseline method can be up to 2.7% and 4.0% respectively.

To further validate the superiority of our AML on point cloud learning, we conduct experiments about the comparison of different margin loss functions including CosFace [19] and ArcFace [20], where the margin is set by the recommendation of the original paper. From Table 4, it can be seen that our AML consistently outperforms the state-of-art CosFace and ArcFace loss.

4.5 Parameters Selection

There are only three free parameters in the proposed method that need to be tuned, *i.e.*, the scaling parameter γ , β and the regularization parameter λ . Figure 3 shows that the classification and retrieval curves as functions of γ , β and λ conducted on ModelNet40. It can be seen

that the performance of the proposed method is insensitive to the rescaling parameter γ , β in generally. In our implementation, γ , β is fixed as 9.6, 0.83, respectively. From Figure 5(c) we can see that the regularization parameter of 10 leads to the best performance, which is adopted in our implementation.

5 Conclusion

In this paper, we propose an adaptive margin-based loss function for point cloud learning, where the margin is adaptively assigned to each class. To further enhance the representation capability of the network, a hierarchical bilinear pooling strategy across layers and a lightweight spatial transform module are also proposed. Extensive experiment results of classification and retrieval show that our method outperforms state-of-art methods based on multiple representations of 3D objects.

Declarations

Funding This work is supported by the Guangxi Key Laboratory of Structure-Activity Relationship of Electronic Information Materials (221035-k) and the National Natural Science Foundation of China (No. 62361015).

References

- [1] Hsu, W.H.: Learning from 3d (point cloud) data. In: Proceedings of the 27th ACM International Conference on Multimedia, pp. 2697–2698 (2019)
- [2] Qin, Z., Wang, J., Lu, Y.: Weakly supervised 3d object detection from point clouds. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 4144–4152 (2020)

- [3] Fu, H., Jia, R., Gao, L., Gong, M., Zhao, B., Maybank, S., Tao, D.: 3d-future: 3d furniture shape with texture. *Int. J. Comput. Vis.* **129**, 3313–3337 (2021)
- [4] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1912–1920 (2015)
- [5] Ma, C., Guo, Y., Yang, J., An, W.: Learning multi-view representation with lstm for 3-d shape recognition and retrieval. *IEEE Trans. Multimed.* **21**, 1169–1182 (2018) <https://doi.org/10.1109/TMM.2018.2875512>
- [6] Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 945–953 (2015)
- [7] Wang, L., Xu, H., Kang, W.: Mvcontrast: Unsupervised pretraining for multi-view 3d object recognition. *Mach. Intell. Res.* **20**, 872–883 (2023) <https://doi.org/10.1007/s11633-023-1430-z>
- [8] Qi, C.R., Su, H., Mo, K., Guibas, L.J.: 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 (2017)
- [9] Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: *Advances in Neural Information Processing Systems*, pp. 5099–5108 (2017)
- [10] Liu, J., Zheng, Y., Yu, R., Wang, Y., Deng, T., Ye, W., Wang, H.: Point Mamba: A Novel Point Cloud Backbone Based on State Space Model with Octree-Based Ordering Strategy. Preprint at <https://arxiv.org/abs/2403.06467v2> (2024)
- [11] He, X., Huang, T., Bai, S., Bai, X.: View n-gram network for 3d object retrieval. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 7515–7524 (2019)
- [12] Huang, S., Cao, J., Dai, Q.: Incremental learning for multi-view 3d shape recognition based on knowledge distillation. In: *2023 4th International Conference on Computer Engineering and Application (ICCEA)*, pp. 19–22 (2023)
- [13] You, H., Feng, Y., Zhao, X., Zou, C., Ji, R., Gao, Y.: Pvrnet: Point-view relation neural network for 3d shape recognition. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 9119–9126 (2019)
- [14] Hegde, V., Zadeh, R.: Fusionnet: 3d object classification using multiple data representations. Preprint at <https://arxiv.org/abs/1607.05695v4> (2016)
- [15] You, H., Ji, R., Feng, Y., Gao, Y.: Pvnet: A joint convolutional network of point cloud and multi-view for 3d shape recognition. In: *Proceedings of the 26th ACM International Conference on Multimedia*, pp. 1310–1318 (2018)
- [16] 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. Graph.* **38**, 1–12 (2019) <https://doi.org/10.1145/3326362>
- [17] Liu, Y., Fan, B., Meng, G., Lu, J., Xiang, S., Pan, C.: Densepoint: Learning densely contextual representation for efficient point cloud processing. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5239–5248 (2019)
- [18] Liu, Y., Fan, B., Xiang, S., Pan, C.: Relation-shape convolutional neural network for point cloud analysis. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8895–8904 (2019)
- [19] Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: Cosface: Large margin cosine loss for deep face recognition. In: *Proceedings of the IEEE Conference on*

- Computer Vision and Pattern Recognition, pp. 5265–5274 (2018)
- [20] Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4690–4699 (2019)
 - [21] Yu, C., Zhao, X., Zheng, Q., Zhang, P., You, X.: Hierarchical bilinear pooling for fine-grained visual recognition. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 574–589 (2018)
 - [22] Qian G, P.H. Li Y: Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In: Advances in Neural Information Processing Systems, vol. 35, pp. 23192–23204 (2022)
 - [23] Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., Wei, Y.: Circle loss: A unified perspective of pair similarity optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6398–6407 (2020)
 - [24] Liu, J., Hou, Q., Cheng, M., Wang, C., Feng, J.: Improving convolutional networks with self-calibrated convolutions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10096–10105 (2020)
 - [25] He, X., Zhou, Y., Zhou, Z., Bai, S., Bai, X.: Triplet-center loss for multi-view 3d object retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1945–1954 (2018)
 - [26] Li, Z., Xu, C., Leng, B.: Angular triplet-center loss for multi-view 3d shape retrieval. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 8682–8689 (2019)
 - [27] Wen, X., Han, Z., Liu, X., Liu, Y.: Lpoint2spatialcapsule: Aggregating features and spatial relationships of local regions on point clouds using spatial-aware capsules. IEEE Trans. Image Process. **29**, 8855–8869 (2020) <https://doi.org/10.1109/TIP.2020.3019925>
 - [28] Liu, Z., Hu, H., Cao, Y., Zhang, Z., Tong, X.: A closer look at local aggregation operators in point cloud analysis. In: European Conference on Computer Vision, pp. 326–342 (2020)
 - [29] Hong, T., Zhang, Z., Ma, J.: Pcsalmix: Gradient saliency-based mix augmentation for point cloud classification. In: 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5 (2023)
 - [30] Savva, M., Yu, F., Su, e.a. H.: Shrec16 track: largescale 3d shape retrieval from shapenet core55. In: Proceedings of the Eurographics Workshop on 3D Object Retrieval, pp. 39–50 (2016)
 - [31] Klovov, R., Lempitsky, V.: Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 863–872 (2017)
 - [32] Bai, S., Bai, X., Zhou, Z., Zhang, Z., Latecki, L.J.: Gift: A real-time and scalable 3d shape search engine. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5023–5032 (2016)
 - [33] Liu, A.-A., Lu, H., Zhou, H., Li, T., Kankanhalli, M.: Balanced class-incremental 3d object classification and retrieval. IEEE Trans. Knowl. Data Eng. **36**, 35–48 (2023) <https://doi.org/10.1109/TKDE.2023.3284032>
 - [34] Sun, H., Wang, Y., Wang, P., Deng, H., Cai, X., Li, D.: Vsformer: Mining correlations in flexible view set for multi-view 3d shape understanding. In: IEEE Transactions on Visualization and Computer Graphics, pp. 2127–2141 (2024)