MBGRLp : Multiscale Bootstrap Graph Representation Learning on Pointcloud (Student Abstract)

Vandan Gorade[1], **Azad Singh**[2], **Deepak Mishra** [2]

[1] University of Pune, Maharashtra - 411007, India
[2] Indian Institute of Technology Jodhpur, Rajasthan - 342037, India
vangorade@gmail.com, singh.63@iitj.ac.in, dmishra@iitj.ac.in

Abstract

Point cloud has gained a lot of attention with the availability of large amount of point cloud data and increasing applications like city planning and self-driving cars. However, current methods, often rely on labeled information and costly processing, such as converting point cloud to voxel. We propose a self-supervised learning approach to tackle these problems, combating labelling and additional memory cost issues. Our proposed method achieves results comparable to supervised and unsupervised baselines on widely used benchmark datasets for self-supervised point cloud classification like ShapeNet, ModelNet10/40.

Introduction

Existing approaches for deep learning on point cloud are based on supervised learning or generative models like GAN and autoencoder (Wu et al. 2017; Yang et al. 2018; Qi et al. 2017a). A couple of attempts have been made for contrastive learning on point cloud (Zhang and Zhu 2019). Still, they mainly depend on a sampling of positive and negative pairs and require point cloud to be converted into voxel or need 2d images of point cloud, which takes additional memory. We propose an approach to directly utilize the irregular point cloud without converting it into voxel.

Our Contributions:

- To the best of our knowledge, we are the first to apply graph contrastive learning on the point cloud data.
- We propose a novel self-supervised learning pretraining approach to combat challenges associated with current supervised approaches for point cloud.
- Through an extensive set of experiments, we validate the proposed method's efficiency and achieve results comparable to supervised techniques designed for point cloud.

Proposed Method

Our architecture is similar to MERIT (Jin et al. 2021), and can be divided into three parts: augmentation, encoder-projector-predictor and multi-scale contrastive learning. Given a point cloud $p = \{x_1, x_2....x_n\}$, where $x_i \in R^d$ and d=3 represents 3D point from set of point clouds P. Our model first produces two augmented views from x and then processes them using online and target encoder which generate representation $Y_\theta = f_\theta(view_1)$, $Z_\theta = g_\theta(Y_\theta)$ and $Y_\epsilon = f_\epsilon(view_2)$, $Z_\epsilon = g_\epsilon(Y_\epsilon)$ and similarly for cross-views. The target encoder outputs Z_1 and Z_2 representation for each view. Online encoder's projected output is then pass to predictor q_θ which generates $H_1 = q_\theta(Z_1)$ and $H_2 = q_\theta(Z_2)$. Now cross-view and cross-model contrastive loss utilizes H_1, H_2 and Z_1, Z_2 while performing multi-scale bootstrapping of representations, similar to MERIT (Jin et al. 2021), as shown in Fig. 1. Once the model is trained we can use learned representation Y_θ for our downstream tasks.

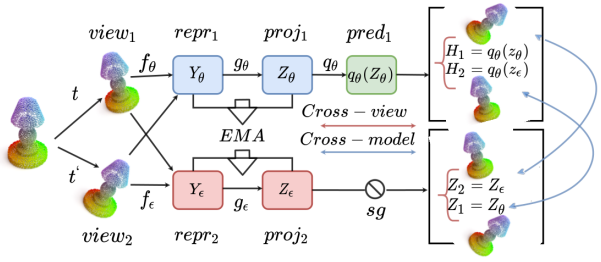

Figure 1: $view_1$ and $view_2$ are two augmented views of input, employed by f_θ and f_ϵ for learning representation r_θ and r_ϵ by utilizing cross-model and cross-view loss. g_θ, g_ϵ and q_θ are the two-layer MLPs. sg is stop gradient.

Augmentation In contrastive learning, selecting the right augmentation is very important. Through our experimental studies we choose the following series of augmentation for each $view$: For $view_1$ we first rotate the input randomly following by jitter, scale and finally shifting. For $view_2$ we first rotate by $45°$ following by jitter and shuffle.

Cross-Model Learning Cross-model contrastive learning uses pairs from two different models, online and target.

$$L^1_{cn}(p_i) = -\log \frac{exp(sim(h^1_{p_i}, z^2_{p_i}))}{\sum_{j=1}^{N} exp(sim(h^1_{p_i}, z^2_{p_j}))}, \qquad (1)$$

In above equation $h^1_{p_i} \in H_1$, $z^2_{p_i} \in Z_1$ and sim(.) denotes cosine similarity. $(h^1_{p_i}, z^2_{p_j})^+$ is the positive pair which attracts two same point-nodes representation from different views of different encoders. The parameters ϵ of target network updates as an exponential moving average of the online encoder parameters θ, i.e. $\epsilon \leftarrow \tau\epsilon + (1-\tau)\theta$. We also construct extra negative pair $(h^1_{p_i}, z^2_{p_j})^-$ which act as regularizer for our loss. Similarly $L^2_{cn}(p_i)$ can be calculated in a similar fashion. By combining $L^1_{cn}(p_i)$ and $L^2_{cn}(p_i)$ we get:

$$L_{cn} = \frac{1}{2N}\sum_{i=1}^{N}(L^1_{cn} + L^2_{cn}) \tag{2}$$

Cross-View Learning Cross-view contrastive learning discriminates the pair representations from two views in the online encoder which acts as strong regularizer to our loss.

$$L^1_{inter}(p_i) = -\log \frac{exp(sim(h^1_{p_i}, h^2_{p_i}))}{\sum_{j=1}^{N} exp(sim(h^1_{p_i}, h^2_{p_j}))}, \tag{3}$$

$L^2_{inter}(p_i)$ for $view_2$ can be obtained in similar fashion.

$$L^1_{intra}(p_i) = -\log \frac{exp(sim(h^2_{p_i}, h^1_{p_i}))}{exp(sim(h^2_{p_i}, h^1_{p_j})) + \gamma}. \tag{4}$$

$L^2_{intra}(p_i)$ for $view_2$ can be obtained in similar fashion. γ denotes cumulative sum of similarity of negative pairs. By combining inter- and intra- loss we get:

$$L_{cv} = \frac{1}{2N}\sum_{i=1}^{N}(L^1_{cv} + L^2_{cv}) \tag{5}$$

End-to-End Learning We combine cross-model and cross-view contrastive loss and defined overall loss as:

$$L = \beta L_{cv} + (1-\beta)L_{cn} \tag{6}$$

Where β is the balance factor. Cross-view and cross-model contrastive routes, act as regularizer and enrich self-supervised signal during optimization.

Experiments and Discussion

We conducted two experiments. First, we performed self-supervised (SS) training on our model on the ShapeNetCore (Chang et al. 2015) dataset, having 55 common object categories with about 51,300 unique 3D models for 150 epochs. In the second experiment, we performed SS-training on our model for 150 epochs on a relatively smaller dataset, ModelNet40, containing 12,311 prealigned shapes from 40 categories. We use only train split (9,843) for the SS-training. We use DGCNN(Wang et al. 2019) as our encoder throught all experiments. For downstream tasks in both experiments, we use ModelNet10 and 40. We use the standard linear evaluation protocol to evaluate our model

Ablation study To understand effect of cross-view and cross-model learning we train our model w/o cross-view and cross-model, and show results in Table 1.

Methods	ModelNet40	ModelNet10
	Linear	Linear
3D-GAN (Wu et al. 2017)	83.3%	91.0%
Latent-GAN (Achlioptas et al. 2018)	85.7%	95.3%
FoldingNet (Yang et al. 2018)	88.4%	94.4%
ContrastNet (Zhang and Zhu 2019)	84.1%	91.0%
PointNet (Qi et al. 2017a)	89.2%	77.6%
PointNet++ (Qi et al. 2017b)	90.7%	-
DGCNN (Wang et al. 2019)	92.9%	-
Ours on S	89.36%	94.28%
Ours on S (10%)	86.75%	90.85%
Ours on M	90.22%	93.30%
Ours on M(10%)	88.17%	91.87%
Ours w/o Cross-view on S	87.43%	92.56%
Ours w/o Cross-model on S	88.70%	93.01%

Table 1: Performance of our model on ModelNet40 and ModelNet10 dataset for Linear evaluation. 'S' and 'M' denotes ShapeNet and ModelNet40 respectively. 10% denotes that only 10% data is used for training of downstream model.

Conclusion

This paper proposed a novel self-supervised learning approach for point cloud and achieved comparable results to supervised techniques.

References

Achlioptas, P.; Diamanti, O.; Mitliagkas, I.; and Guibas, L. 2018. Learning Representations and Generative Models for 3D Point Clouds. arXiv:1707.02392.

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; Xiao, J.; Yi, L.; and Yu, F. 2015. ShapeNet: An Information-Rich 3D Model Repository. arXiv:1512.03012.

Jin, M.; Zheng, Y.; Li, Y.-F.; Gong, C.; Zhou, C.; and Pan, S. 2021. Multi-Scale Contrastive Siamese Networks for Self-Supervised Graph Representation Learning. arXiv:2105.05682.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. arXiv:1612.00593.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. arXiv:1706.02413.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic Graph CNN for Learning on Point Clouds. arXiv:1801.07829.

Wu, J.; Zhang, C.; Xue, T.; Freeman, W. T.; and Tenenbaum, J. B. 2017. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. arXiv:1610.07584.

Yang, Y.; Feng, C.; Shen, Y.; and Tian, D. 2018. FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation. arXiv:1712.07262.

Zhang, L.; and Zhu, Z. 2019. Unsupervised Feature Learning for Point Cloud by Contrasting and Clustering With Graph Convolutional Neural Network. arXiv:1904.12359.