Self-Supervision for 3D Real-World Challenges

Antonio Alliegro[1], Davide Boscaini[2], and Tatiana Tommasi[1]

[1]Politecnico di Torino, Turin, Italy {name.surname}@polito.it
[2]Fondazione Bruno Kessler, Trento, Italy dboscaini@fbk.eu

Abstract. We consider several possible scenarios involving synthetic and real-world point clouds where supervised learning fails due to data scarcity and large domain gaps. We propose to enrich standard feature representations by leveraging self-supervision through a multi-task model that can solve a 3D puzzle while learning the main task of shape classification or part segmentation.

1 Introduction

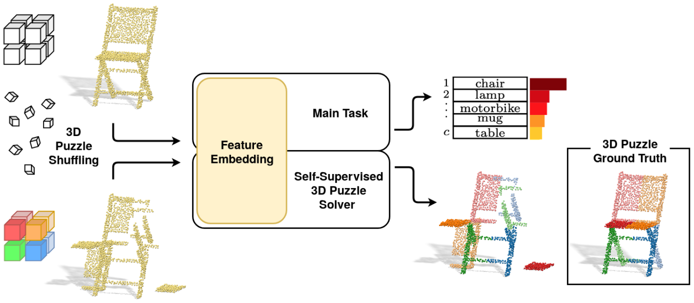
Point clouds are the standard representation for 3D data, but they come with three main drawbacks: they are un-structured, un-ordered and eager for precise manual annotation due to many possible sources of noise. The first two issues make typical convolutional neural networks (CNN) unsuitable, while the third has initially guided research towards very well lab-controlled and synthetic CAD object datasets where labeling is simpler. The most recent results on those testbed are witnessing a trend of performance saturation raising the question of how to move forward. Self-supervised learning is helpful in this respect: a simple task like solving a 3D puzzle leverages on the spatial co-location of shape parts and exploits reliable knowledge on relative point positions at global and local level.

2 Method

We propose a *new multi-task end-to-end deep learning model for point clouds that combines supervised and self-supervised learning* (see Fig. 1). Specifically, we build on top of PointNet [5] and PointNet++ [6] backbones a deep architecture that solves 3D puzzles while jointly training a main supervised task. We show how these two tasks complement each other making the obtained model (a) more robust in case of scarce labeled data, (b) easier to transfer for adaptation and (c) more reliable for out of domain generalization. By extensive experiments across three different point clouds datasets we show that our multi-task method defines the new state-of-the-art for both shape classification and part segmentation in the most challenging real world settings.

More formally Our multi-task model can be described as the combination of two parametric non-linear functions: $\mathbf{\Phi}_{\theta_f,\theta_m}$ and $\mathbf{\Psi}_{\theta_f,\theta_p}$, where the subscripts of the parameters θ refer respectively to the feature extraction (f), main task (m), and

Fig. 1. Overview of the proposed multi-task approach



puzzle solution (p) modules of our deep network. The feature encoder is shared between the two functions. For each sample $\mathbf{x}$ that enters the network, $\mathbf{\Phi}_{\theta_f,\theta_m}(\mathbf{x})$ is the output of the feature extractor and final fully connected part of the network. The loss function $\mathcal{L}_m(\mathbf{\Phi}_{\theta_f,\theta_m}(\mathbf{x}),\mathbf{y})$ measures the prediction error on the main task. The auxiliary function $\mathbf{\Psi}$ deals with a *puzzled* variant $\tilde{\mathbf{x}} = \mathcal{P}(\mathbf{x})$ of the original input point cloud. To get it, we start from $\mathbf{x}$, scale it to unit cube and split each axis into $l = 3$ equal lengths intervals forming l^3 voxels which are labeled according to their original position. Each vertex contained inside a voxel inherits its label. Finally, all the voxels are randomly swapped, producing a new shuffled point cloud. We indicate with $\tilde{S} = \{(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^N$ the obtained puzzled samples where the voxel position label for each point is $y_{ik} \in \{1,\ldots,l^3\}$. Once these new displaced data are encoded in the feature latent space, a second network head focuses on solving the 3D puzzle problem by minimizing the auxiliary loss that measures the reordering error $\mathcal{L}_p(\mathbf{\Psi}_{\theta_f,\theta_p}(\tilde{\mathbf{x}}),\tilde{\mathbf{y}})$ in terms of difference between the assigned voxel label and the correct one per point. The training objective is:

$$\arg\min_{\theta_f,\theta_m,\theta_p} \sum_{i=1}^{N} \mathcal{L}_m(\mathbf{\Phi}_{\theta_f,\theta_m}(\mathbf{x}_i),\mathbf{y}_i) + \alpha\,\mathcal{L}_p(\mathbf{\Psi}_{\theta_f,\theta_p}(\tilde{\mathbf{x}}_i),\tilde{\mathbf{y}}_i), \qquad (1)$$

where both $\mathcal{L}_m$ and $\mathcal{L}_p$ are cross-entropy losses. Note that, while the first loss deals only with original samples, the second involve both original and puzzled samples, given the random nature of the voxel shuffling procedure. The described learning problem has one main hyper-parameter α, which weights the self-supervised loss, we set $\alpha = 0.6$ for all our analysis.

3 Experiments on Cross-Domain Classification

We evaluate the cross-domain classification performance of our multitask on synthetic and Real-World data respectively from ModelNet40 [10] and ScanObjectNN [8]. The latter offering several splits of the same data with increasing difficulty (`OBJ_ONLY` to `PB_T50_RS_BG`) in terms of background, noise, shape perturbation. *Baselines.* We use as reference the standard supervised baseline. It is a naïve

Table 1. Shape classification accuracy (%) when training and testing is done on different domains (DG). If the unlabeled target data is provided at training time (DA), our multi-task is able to adapt and reduce the domain gap

Classification - Domain Generalization and Adaptation

Method		ModelNet40 →					PB_T50_RS_BG →
		OBJ_ONLY	OBJ_BG	PB_T50_RS	PB_T50_RS_BG	AVG	ModelNet40
PointDAN [7]		56.42	44.84	48.99	34.39	46.16	54.66
PN	Baseline	54.74	43.58	44.96	34.25	44.38	47.43
	Our DG	54.53	49.68	45.22	36.28	46.43	39.30
	Our DA	58.53	47.58	46.70	35.85	47.16	51.54
PN++	Baseline	52.49	44.00	44.83	34.29	43.90	47.66
	Our DG	57.47	52.42	52.84	38.65	50.34	52.88
	Our DA	60.4	53.89	54.66	39.63	**52.14**	**56.07**
3DmFV [1]		30.90	24.00	24.90	16.40	24.05	51.50
PointCNN [4]		32.20	29.50	24.60	19.20	26.37	49.20
DGCNN [9]		49.30	46.70	36.80	27.20	40.00	54.70

variant of our method obtained by turning off the puzzle solver ($\alpha = 0$ in Eq. 1).

Domain Generalization. When training and test data are drawn from two very different distributions the model learned on the former one usually fails to generalize to the latter. We consider the DG setting when training on ModelNet40 and testing on ScanObjectNN and report results in Table 1. Our multi-task approach fully trained on only synthetic data shows a significant improvement with respect to the baseline with gains up to 6 and 8 pp in the OBJ_BG and with a still relevant gain of 2 and 4 pp in the most challenging PB_T50_RS_BG, respectively with PN and PN++ encoders. We also consider the inverse generalization direction from PB_T50_RS_BG to ModelNet40 with compelling results.

Unsupervised Domain Adaptation. We also investigated whether our multi-task approach could close the domain gap when unlabeled target data are available at training time, given its unsupervised nature these data are fed to our puzzle solver. DA results in Table 1 provide a positive answer showing a further increase in performance over the DG results. The recent PointDAN method [7] proposed to solve point cloud domain shifts by combining local nodes alignment and global features alignment. Table 1 shows that our multi-task approach largely outperforms this solution. Finally, an overall look at the performance of several recent point cloud networks is provided in the bottom part of Table 1: our multi-task approach establishes the new state-of-the-art.

4 Experiments on Part Segmentation

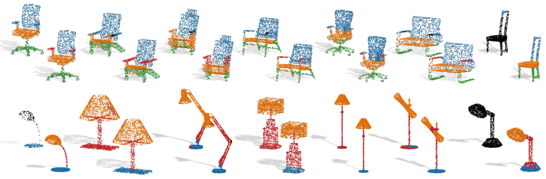
We focus on the case of scarce labeled data availability when dealing with part segmentation. The quality of the predicted part segmentation is evaluated in terms of the mean Intersection-over-Union (mIoU) metric.

Few-Shot and Semi-Supervised. By following [2] we randomly sample 1% and 5% of the ShapeNetPart train set to evaluate the point features in a semi-supervised setting. The results in Table 2 indicate that our multi-task approach, although not

Table 2. Accuracy (mIoU) for part segmentation on ShapeNetPart with limited annotations

Method	1%	5%
SO-Net [3]	64.00	69.00
PointCapsNet [11]	67.00	70.00
CCD [2]	68.20	**77.70**
Baseline	64.52	75.75
Our FS	64.49	75.07
Our SS	**71.95**	77.42

Fig. 2. Part segmentation of chairs and lamps when 1% of training data are available. Baseline prediction (top left) and our approach (bottom right). Black points denotes predictions whose maximum value was not a chair or lamp part



improving over the baseline in the few-shot setting, in the semi-supervised setting outperforms the current state of the art in the case of only 1% of supervised data while practically matches it in the 5% case. We plot some visualizations out of our 1% part segmentation experiment in Figure 2 for chairs and lamps. Regarding chairs, our multi-task approach seems to allow a better recognition of the armrests. Indeed the position of these relative small parts of the chair may be better learned thanks to the auxiliary puzzle solution task. A similar consideration may be done for the lamp basis.

References

1. Ben-Shabat, Y., Lindenbaum, M., Fischer, A.: 3dmfv: Three-dimensional point cloud classification in real-time using convolutional neural networks. IEEE RA-L (2018)
2. Hassani, K., Haley, M.: Unsupervised multi-task feature learning on point clouds. In: ICCV (2019)
3. Li, J., Chen, B.M., Hee Lee, G.: So-net: Self-organizing network for point cloud analysis. In: CVPR (2018)
4. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: Convolution on x-transformed points. In: NIPS (2018)
5. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In: CVPR (2017)
6. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In: NIPS (2017)
7. Qin, C., You, H., Wang, L., Kuo, C.C.J., Fu, Y.: Pointdan: A multi-scale 3d domain adaption network for point cloud representation. In: NIPS (2019)
8. Uy, M.A., Pham, Q.H., Hua, B.S., Nguyen, D.T., Yeung, S.K.: Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data. In: ICCV (2019)
9. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic Graph CNN for Learning on Point Clouds. TOG (2019)
10. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3D ShapeNets: A Deep Representation for Volumetric Shapes. In: CVPR (2015)
11. Zhao, Y., Birdal, T., Deng, H., Tombari, F.: 3d point capsule networks. In: CVPR (2019)