http://eprints.gla.ac.uk/268682/

Deposited on 11 April 2022

Towards Robust 3D Object Recognition with Dense-to-Sparse Deep Domain Adaptation

Prajval Kumar Murali[*‡], Cong Wang[*†], Ravinder Dahiya[‡], and Mohsen Kaboli[*§]

[*]BMW Group, München, Germany, [†]Technical University of Munich, Germany.
[‡]University of Glasgow, Scotland, [§]Radboud University, Netherlands.
Email: prajval-kumar.murali@bmwgroup.com, ge57qom@mytum.de,
ravinder.dahiya@glasgow.ac.uk, mohsen.kaboli@bmwgroup.com

Abstract—Three-dimensional (3D) object recognition is crucial for intelligent autonomous agents such as autonomous vehicles and robots alike to operate effectively in unstructured environments. Most state-of-art approaches rely on relatively dense point clouds and performance drops significantly for sparse point clouds. Unsupervised domain adaption allows to minimise the discrepancy between dense and sparse point clouds with minimal unlabelled sparse point clouds, thereby saving additional sparse data collection, annotation and retraining costs. In this work, we propose a novel method for point cloud based object recognition with competitive performance with state-of-art methods on dense and sparse point clouds while being trained only with dense point clouds.

Index Terms—Deep Domain Adaptation, Point Cloud Classification, Object Recognition, Deep Neural Networks

I. INTRODUCTION

Point cloud based approaches are becoming popular for various applications in the automotive and robotics domain such as object recognition, scene understanding, pose estimation and so on due to the ubiquity of 3D visual sensors. Typical visual sensors such as RGB-D cameras produce around 10^4 points with a single snapshot. However, LiDAR-based point clouds are very sparse with increasing distance, ranging between 10-100 points [1], [2]. Such sparse point clouds are also produced with commercial tactile sensors [3]–[5] (Figure 1). Most state-of-art methodologies rely on relatively dense point clouds and the performance drops drastically with sparse point clouds [6]. In particular case for tactile sensing, acquiring larger number of point clouds is directly proportional to the number of contact actions performed using the tactile sensor [7]–[12]. Hence, for increased robustness and efficiency of autonomous agents (robots or autonomous vehicles) it is imperative for perception algorithms to perform adequately with limited amount of data availability.

Recently, deep learning approaches have been proposed for working with raw 3D point clouds for various tasks such as semantic segmentation, object pose estimation and object recognition [14]. However, it is challenging due to the unstructured nature of point clouds, small-scale datasets and high dimensionality of the data [14]. PointNet [15] and PointNet++ [16] were the first works in this domain and several derivative networks have been proposed based on the PointNet architecture [14]. According to various studies, the performance of such networks degrades dramatically when
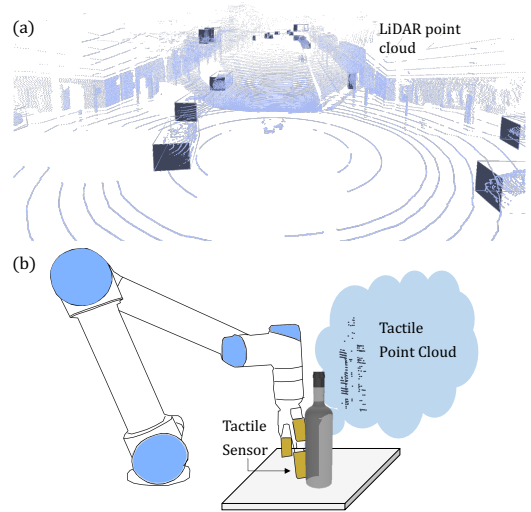

Fig. 1: Typical applications with sparse point clouds: (a) LiDAR scans in autonomous vehicles for object/ pedestrian detection and localisation (from PandaSet [13]); (b) robots with tactile sensors performing object recognition and manipulation.

sparse point clouds with point numbers ranging from 10 to 100 are used [6], [16]–[18]. The sparse point clouds pose a significant challenge due to inability to extract meaningful features such as surface normals and local curvature. Even for humans, it becomes increasingly difficult to discern the point clouds for identifying objects when the number of points falls below 100 [6]. Using the ModelNet40 dataset [19], Xiao et al. [6] introduced Triangle-Net which is trained on dense and sparse point clouds and demonstrates a classification accuracy of 70% using less than 20 points. However, retraining large networks with sparse point cloud data adds additional overhead costs such as data acquisition, data labelling and training. Domain adaptation techniques can help bridge the gap by using the trained networks on dense point clouds that are relatively easily available with a smaller subset of unlabelled sparse point clouds. This helps reduce data collection and labelling costs by a human annotator while increasing model robustness by allowing the models to work on various types of input data. There are several approaches for unsupervised
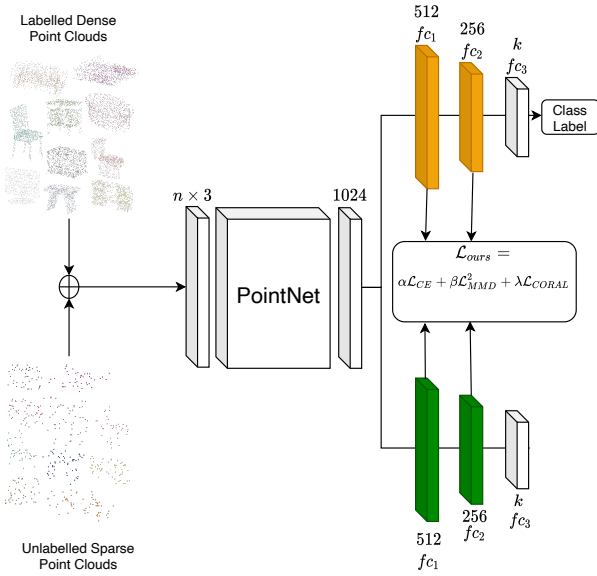
Fig. 2: Network architecture for dense to sparse unsupervised domain adaptation for object recognition

domain adaptation available in literature [20], [21] wherein we concentrate on discrepancy based techniques such as Maximum mean discrepancy (MMD) [22] and Correlation Alignment (CORAL) [23]. The discrepancy-based methods works by reducing the distance between the source and target domains using statistic criteria.

Contributions:

In this work, we propose a novel unsupervised domain adaptation method from dense to sparse point clouds for object recognition. Our network is trained with dense labelled point clouds and adapted with a small subset of unlabelled sparse point clouds. We use the ModelNet10 [19] dataset for evaluating our proposed method. Furthermore, we benchmark our approach against other state-of-art point cloud classification methods such as Triangle-Net [6], PointNet [15], and DGCNN [24].

II. METHODOLOGY

As the target domain has no labelled data, it is not possible to fine-tune the network that is trained on the source domain data to the target domain. Hence, we train our network on the labelled source dataset and use discrepancy methods of domain adaptation to adapt the trained model with the unlabelled target dataset. Our network architecture is shown in Figure 2. The input to the network is n point clouds with x, y, z data associated with each point. The PointNet [15] architecture is used for extracting the features from the raw point cloud data. The network provides as output k classification scores for each of the k classes. We get a global feature vector of size 1024 from the PointNet output which is used by a series of fully connected layers for the classification task. We use 3 fully connected layers fc of sizes $512, 256, k$. We trained our network with dense point clouds from ModelNet10 [19] wherein the point clouds of size 1024 points are sampled

randomly from the CAD meshes in ModelNet10 [19]. The output size is $k = 10$ as there are 10 classes of objects in ModelNet10 [19]. We use the cross-entropy loss for training. This represents the source dataset $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ with n_s labelled dense point cloud samples. The target domain is represented as $D_t = \{x_j^t\}_{j=1}^{n_t}$ with n_t unlabelled samples comprising of sparse point clouds of 50 points sampled from the ModelNet10 dataset. We employ discrepancy-based methods to extract domain-invariant representations for the unsupervised domain adaptation. Maximum Mean Discrepancy (MMD) [22] and Correlation Alignment (CORAL) [23] are well-known methods for domain adaptation that has been extensively applied in the visual domain. Given labelled source domain $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ and unlabelled target domain $D_t = \{x_j^t\}_{j=1}^{n_t}$ with probability distributions p^s and p^t respectively, the MMD between p^s and p^t is defined as:

$$\text{MMD}^2(p^s, p^t) = \sup_{||\phi||_{\mathcal{H}} \leq 1} ||\mathbb{E}_{x^s \sim p^s}[\phi(x^s)] - \mathbb{E}_{x^t \sim p^t}[\phi(x^t)]||_{\mathcal{H}}^2,$$
(1)

where $\mathcal{H}$ is the reproducing kernel Hilbert space (RKHS), $\phi(.)$ is the feature mapping associated with the kernel map $k(x^s, x^t) = < \phi(x^s), \phi(x^t) >$, sup(.) is the supremum of the input aggregate and $||\phi||_{\mathcal{H}} \leq 1$ defines a set of functions in the unit ball of $\mathcal{H}$. The multi-kernel MMD (MK-MMD) [25] obtains the optimal kernel as a linear combination of individual kernels. The kernel $k(x^s, x^t)$ is defined as the convex combination of b positive semi-definite kernels $\{k_u\}$ [25]:

$$K \triangleq \left\{ k = \sum_{u=1}^{b} \beta_u k_u : \sum_{u=1}^{b} \beta_u = 1, \beta_u \geq 0, \forall u \right\},$$
(2)

wherein k is weighted by the different kernels and the coefficients β_u are the weights to ensure that the generated multi-kernel k is characteristic. MK-MMD compares all orders of statistics between the source and target domain while in comparison, CORAL [23] compares the 2nd order statistics of the source and target distributions. Sun et al. [26] proposed Deep-CORAL that uses CORAL for deep neural networks and is defined as follows:

$$\text{CORAL}(x^s, x^t) = \frac{1}{4d^2} ||C_s - C_t||_F^2,$$
(3)

where $||.||_F^2$ is the squared matrix Frobenius norm, C_s and C_t are the covariance matrices of the source and target domain data. Our loss function is defined as the weighted linear combination of MK-MMD, CORAL and the classification loss for the source dataset. Furthermore, previous research has demonstrated that modifying a single layer alone is insufficient to correct the dataset bias between the source and target domains caused by non-transferable layers [27]. Hence, we also use fully-connected layers fc_1 and fc_2 for multi-layer domain adaptation. Our proposed loss function $\mathcal{L}_{ours}$ is:

$$\mathcal{L}_{ours} = \alpha \mathcal{L}_{crossEnt} + \beta \{\mathcal{L}_{MK-MMD}^2\}_{fc_1} + \beta \{\mathcal{L}_{MK-MMD}^2\}_{fc_2} + \lambda \{\mathcal{L}_{CORAL}\}_{fc_1} + \lambda \{\mathcal{L}_{CORAL}\}_{fc_2},$$
(4)

TABLE I: Comparison results against state-of-art methods

Points Num	1024	50	40	30	20	10
PointNet [15]	92.32%	81.03%	76.86%	68.09%	58.55%	32.89%
TriangleNet [6]	81.36%	64.45%	59.89%	47.97%	33.79%	11.06%
DGCNN [24]	93.27%	3.69%	3.12%	2.03%	0.89%	N.A
Our Model	91.12%	82.35%	81.69%	74.89%	65.58%	41.18%

TABLE II: Test Result on ModelNet10

Points Num	$\mathcal{L}_{CORAL}$	$\mathcal{L}_{MMD}$	$\mathcal{L}_{ours}$
1024	91.13%	91.09%	91.12%
512	90.68%	91.19%	90.79%
256	90.35%	90.35%	91.19%
128	91.12%	89.25%	89.80%
64	85.96%	84.98%	86.18%
50	83.88%	82.79%	82.35%
40	80.15%	75.88%	81.69%
30	74.23%	73.90%	74.89%
20	64.03%	58.66%	64.58%
10	40.90%	38.05%	41.18%

TABLE III: Test set accuracy with $\mathcal{L}_{CORAL}$ over various α ($\lambda = 0.5$)

Points Num	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 5$	$\alpha = 10$
1024	90.35%	90.46%	89.91%	90.57%	90.57%
512	90.24%	91.23%	90.24%	91.23%	90.57%
256	89.69%	90.24%	89.25%	90.02%	90.35%
128	88.05%	89.80%	87.39%	88.60%	88.27%
64	82.68%	84.32%	83.11%	82.79%	85.42%
50	81.14%	80.04%	78.51%	78.50%	82.02%
40	77.30%	77.19%	75.33%	73.14%	78.29%
30	72.15%	70.29%	67.98%	67.11%	72.04%
20	58.77%	56.80%	56.69%	56.91%	63.60%
10	37.28%	35.53%	32.67%	35.75%	39.80%

TABLE IV: Test set accuracy with $\mathcal{L}_{ours}$ over various β ($\alpha = 10, \lambda = 0.5$)

Points Num	$\beta = 0.1$	$\beta = 0.5$	$\beta = 5$	$\beta = 10$
1024	91.23%	91.12%	90.24%	90.13%
512	91.67%	90.79%	90.90%	90.79%
256	90.68%	91.19%	90.46%	89.80%
128	89.80%	89.80%	88.38%	89.04%
64	85.11%	86.18%	83.33%	84.87%
50	82.68%	82.35%	78.73%	80.70%
40	79.28%	81.69%	75.77%	77.52%
30	72.92%	74.89%	69.85%	69.85%
20	59.65%	64.58%	56.69%	59.10%
10	36.51%	41.18%	37.50%	34.43%

where α, β, λ are hyperparameters that need to be tuned empirically. The model is adapted to the target domain by minimising the loss $\mathcal{L}_{ours}$.

III. EXPERIMENTAL RESULTS

We used the ModelNet10 dataset containing 4,899 pre-aligned shapes from 10 categories. The training and domain adaptation of the network was performed using PyTorch framework on a workstation with NVidia Quadro RTX 4000 GPU with 8 GB RAM. The training set consists of two parts: supervised training with labelled dense point clouds and unsupervised domain adaptation with unlabelled sparse point clouds. The labelled dense point clouds are generated by randomly subsampling 3,991 (80%) objects in ModelNet10 to 1024 points. The unlabeled sparse point clouds are generated by randomly subsampling the same samples with random reshuffling to 50 points. The test set is generated by randomly subsampling 908 (20%) objects to 50 points. The dataset is further augmented by randomly rotating each sample around the z axis for rotation invariance. The network is first trained by labeled dense point clouds with 150 epochs. Then, the PointNet layers are frozen and the network is trained based on $\mathcal{L}_{ours}$ in Equation(4) with unlabelled sparse data. Table I shows the overall results against other state-of-art methods such as Triangle-Net [6], PointNet [15], and DGCNN [24]. The other state-of-art methods in Table I are solely trained with dense point cloud dataset without domain adaptation. Table II shows the accuracy on the test set when using $\mathcal{L}_{MMD}$, $\mathcal{L}_{CORAL}$ and the $\mathcal{L}_{ours}$ loss in training as ablation studies for different number of points. Table III shows test accuracy results over different α, and Table IV shows the results over different β for hyper-parameter tuning.

As can be seen from Table I, other point cloud-based classification methods trained on dense point cloud perform poorly with low number of points. Our method with domain adaptation using unlabelled sparse clouds of 50 points helps in increasing classification accuracies even as low as 10 points.

This further emphasises the fact that popular 3D classification networks are not robust to point density. Furthermore, we see that our proposed loss L_{ours} performs favourably in comparison to L_{CORAL} and L_{MMD}. The marginal improvement may arise from the fact that the dense and sparse dataset is sampled from the same CAD mesh. We must note that our method involves tuning various hyper-parameters such as α, β and λ as in Table III,IV but provides higher flexibility wherein we can provide more importance to source data (in case of noisy target data) or match certain distributional statistics (in case of *a priori* known correlation) between the domains by tuning our hyperparameters.

IV. CONCLUSION

In this work, we tackled the problem of sparse point cloud based object recognition by using unsupervised domain adaptation with dense labelled point clouds and unlabelled sparse point clouds. We proposed a novel domain adaptation loss by combining MMD loss and CORAL loss in a weighted linear combination. We performed extensive experiments on ModelNet10 [19] with benchmark experiments against state-of-art methods. Our method allows for increased robustness of perceptual pipelines against various types of input data for autonomous agents with minimal overhead costs for data collection and annotation. Future work will focus on extending the proposed methodology to real visuo-tactile data from novel tactile sensors [28]–[30] as well as extend to other contact rich information provided by tactile sensing such as texture, temperature and so on [31]–[34].

References

[1] L. Wang and B. Goldluecke, "Sparse-pointnet: See further in autonomous vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7049–7056, 2021.

[2] P. K. Murali, M. Kaboli, and R. Dahiya, "Intelligent in-vehicle interaction technologies," *Advanced Intelligent Systems*, p. 2100122, 2021.

[3] Q. Li *et al.*, "A review of tactile information: Perception and action through touch," *IEEE Trans. on Rob.*, vol. 36, no. 6, pp. 1619–1634, 2020.

[4] P. K. Murali, M. Gentner, and M. Kaboli, "Active visuo-tactile point cloud registration for accurate pose estimation of objects in an unknown workspace," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 2838–2844.

[5] P. K. Murali, A. Dutta, M. Gentner, E. Burdet, R. Dahiya, and M. Kaboli, "Active visuo-tactile interactive robotic perception for accurate object pose estimation in dense clutter," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4686–4693, 2022.

[6] C. Xiao and J. Wachs, "Triangle-net: Towards robustness in point cloud learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 826–835.

[7] M. Kaboli *et al.*, "Tactile-based active object discrimination and target object search in an unknown workspace," *Autonomous Robots*, vol. 43, no. 1, pp. 123–152, 2019.

[8] M. Kaboli and G. Cheng, "Robust tactile descriptors for discriminating objects from textural properties via artificial robotic skin," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 985–1003, 2018.

[9] M. Kaboli, D. Feng, and G. Cheng, "Active tactile transfer learning for object discrimination in an unstructured environment using multimodal robotic skin," *International Journal of Humanoid Robotics*, vol. 15, no. 01, p. 1850001, 2018.

[10] M. Kaboli, K. Yao, and G. Cheng, "Tactile-based manipulation of deformable objects with dynamic center of mass," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 752–757.

[11] M. Kaboli, D. Feng, K. Yao, P. Lanillos, and G. Cheng, "A tactile-based framework for active object learning and discrimination using multimodal robotic skin," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2143–2150, 2017.

[12] P. K. Murali, R. Dahiya, and M. Kaboli, "An empirical evaluation of various information gain criteria for active tactile action selection for pose estimation," in *The IEEE Int. Conf on Flexible and Printable Sensors and Systems (FLEPS 2022)*. IEEE, 2022.

[13] P. Xiao, Z. Shao, S. Hao, Z. Zhang, X. Chai, J. Jiao, Z. Li, J. Wu, K. Sun, K. Jiang *et al.*, "Pandaset: Advanced sensor suite dataset for autonomous driving," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 3095–3101.

[14] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3d point clouds: A survey," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[15] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[16] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[17] D. Stutz and A. Geiger, "Learning 3d shape completion under weak supervision," *International Journal of Computer Vision*, vol. 128, no. 5, pp. 1162–1181, 2020.

[18] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8895–8904.

[19] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[20] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 5, pp. 1–46, 2020.

[21] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[22] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006.

[23] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[24] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *AAAI*, 2018, pp. 4438–4445.

[25] A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B. K. Sriperumbudur, "Optimal kernel choice for large-scale two-sample tests," in *Advances in neural information processing systems*. Citeseer, 2012, pp. 1205–1213.

[26] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *European conference on computer vision*. Springer, 2016, pp. 443–450.

[27] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in Neural Information Processing Systems*, vol. 27, pp. 3320–3328, 2014.

[28] O. Ozioko, P. Karipoth, P. Escobedo, M. Ntagios, A. Pullanchiyodan, and R. Dahiya, "Sensact: The soft and squishy tactile sensor with integrated flexible actuator," *Advanced Intelligent Systems*, vol. 3, no. 3, p. 1900145, 2021.

[29] P. Escobedo, M. Ntagios, D. Shakthivel, W. T. Navaraj, and R. Dahiya, "Energy generating electronic skin with intrinsic tactile sensing without touch sensors," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 683–690, 2020.

[30] M. Ntagios, H. Nassar, A. Pullanchiyodan, W. T. Navaraj, and R. Dahiya, "Robotic hands with intrinsic tactile sensing via 3d printed soft pressure sensors," *Advanced Intelligent Systems*, vol. 2, no. 6, p. 1900080, 2020.

[31] Y. Kumaresan, O. Ozioko, and R. Dahiya, "Multifunctional electronic skin with a stack of temperature and pressure sensor arrays," *IEEE Sensors Journal*, vol. 21, no. 23, pp. 26 243–26 251, 2021.

[32] M. Soni, M. Bhattacharjee, M. Ntagios, and R. Dahiya, "Printed temperature sensor based on pedot: Pss-graphene oxide composite," *IEEE Sensors Journal*, vol. 20, no. 14, pp. 7525–7531, 2020.

[33] R. Dahiya, N. Yogeswaran, F. Liu, L. Manjakkal, E. Burdet, V. Hayward, and H. Jörntell, "Large-area soft e-skin: The challenges beyond sensor designs," *Proceedings of the IEEE*, vol. 107, no. 10, pp. 2016–2033, 2019.

[34] R. Dahiya, "E-skin: from humanoids to humans," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 247–252, 2019.