2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506426

POINTVIEW-GCN: 3D SHAPE CLASSIFICATION WITH MULTI-VIEW POINT CLOUDS

Seyed Saber Mohammadi[†‡] *Yiming Wang*[†*] *Alessio Del Bue*[⋆†]

[⋆] Pattern Analysis and Computer Vision, Istitalino di Tecnologia, Genova, Italy.
[†]Visual Geometry and Modelling (VGM), Istituto Italiano di Tecnologia, Genova, Italy.
[‡] Department of Electrical, Electronics and Telecommunication Engineering,
Universita degli Studi di Genova, Genova, Italy.
[*]Deep Visual Learning (DVL), Fondazione Bruno Kessler, Trento, Italy.

ABSTRACT

We address 3D shape classification with partial point cloud inputs captured from multiple viewpoints around the object. Different from existing methods that perform classification on the complete point cloud by first registering multi-view capturing, we propose PointView-GCN with multi-level Graph Convolutional Networks (GCNs) to hierarchically aggregate the shape features of single-view point clouds, in order to encode both the geometrical cues of an object and their multi-view relations. With experiments on our novel single-view datasets, we prove that PointView-GCN produces a more descriptive global shape feature which stably improves the classification accuracy by $\sim 5\%$ compared to the classifiers with single-view point clouds, and outperforms the state-of-the-art methods with the complete point clouds on ModelNet40.

Index Terms— Feature Aggregation, Graph Convolutional Network, 3D Shape Classification, Dataset

1. INTRODUCTION

The problem of recognising objects in 3D has recently gained increasing research attention due to its essential role in many real-world applications, such as autonomous driving and domestic robots. With progresses made by the seminal works, PointNet [1] and its follow-up PointNet++ [2], recent works are able to perform analysis directly on unordered point cloud data (PCD) with the advances of deep learning techniques [3].

When performing 3D shape recognition, most of the works [4, 2, 5, 6, 7] take as input the complete scan of an object, which in real world is often obtained by reconstructing the object in 3D with partial PCDs captured from multiple views, as shown in Fig. 1. Instead of performing such PCD-level integration for shape recognition [8], we propose *PointView-GCN*, to aggregate the shape features from the multi-view partial PCDs in order to further exploit the relations among multiple views in the feature space. With experiments on both synthetic [9] and real-world datasets [10], we prove that *PointView-GCN* is able to produce a

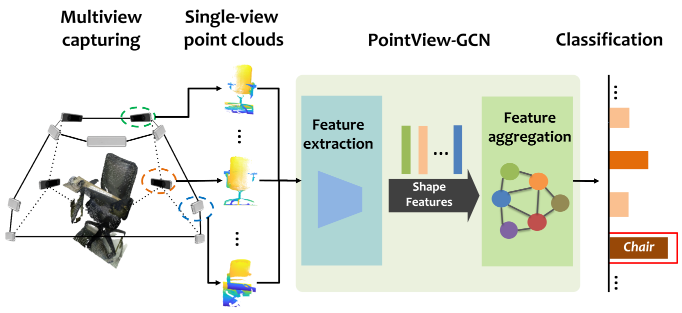

Fig. 1: Overall pipeline of *PointView-GCN*, where multi-view partial point clouds of an object are fed to a backbone network to extract their shape features, forming a view-graph. We then apply a multi-level GCN on the view-graph to aggregate features for shape classification.

more descriptive global shape feature for the shape recognition task, outperforming the state-of-the-art (sota) works that take the complete PCD as input. There exist various methods for multi-view feature aggregation with image data [11, 12], where Graph Convolutional Network (GCN) has demonstrated its strength for feature aggregation in terms of encoding the semantic relations among multiple views [13].

Our method *PointView-GCN* proposes a hierarchical network with multi-level GCNs to aggregate the shape features from partial PCDs captured from multiple views, exploiting the semantic relations among neighbouring views in a fine-to-coarse manner. Our main contributions can be summarised as: 1) We propose *PointView-GCN* with multi-level GCNs for multi-view shape feature aggregation. The aggregated feature is more descriptive and outperforms the sota PCD-based methods on the benchmark dataset ModelNet40 [9]. 2) We propose a skip connection across different hierarchies of the GCN, where the structure of the hierarchy has also been optimally chosen with evidence of extensive experiments. 3) We propose novel datasets with partial single-view PCDs generated from benchmark datasets [9, 10]. Code and datasets is

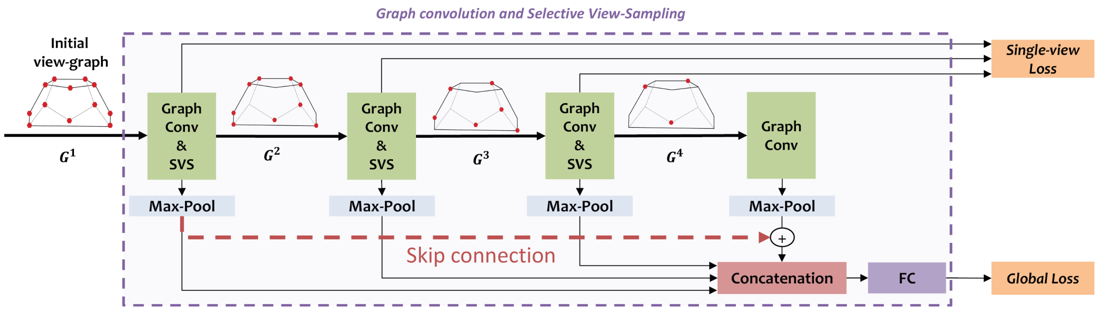

Fig. 2: Feature aggregation of *PointView-GCN* using four levels of GCNs with the input view graph G_1, whose node is represented by the shape feature extracted from each single-view PCD. Each level of the GCN first updates the node by aggregating the neighbourhood information, and then forms a sub view-graph via selective view-sampling (SVS). A skip connection between the global feature at the first and the last level is introduced to improve the classification performance.

available on our official repository[1].

2. RELATED WORK

We mainly cover related works addressing the shape classification with 3D data and with multi-view inputs.

Shape classification with 3D data. There exist various approaches addressing the shape classification problem with different 3D representations. Voxel-based CNN [3] shows promising performance however expensive in memory usage. PointNet[1] was proposed to directly process PCD with a network for shape analysis. However, since PointNet does not capture local structures, therefore has limitations in recognition of fine-grained patterns. The follow-up work PointNet++ [2] addresses this limitation by processing a set of points sampled in a metric space in a hierarchical fashion. Differently, DGCNN [5], proposes a simple operation, named EdgeConv, to capture local geometric structure by generating edge features describing the relationships among a point and its neighbours. The above-mentioned works tackles the complete PCD of objects, while our work focuses on aggregating shape features from partial PCDs captured from multiple views: the latter is the typical setting in which 3D data are acquired in practical scenarios.

Multi-view 3D shape classification. MVCNN [14] first proposes to apply 2D CNNs for recognising 3D object using max-pooling to aggregate features from different views, in order to achieve a global shape descriptor. However, this method does not consider the semantic relations among multi-view data during feature aggregation [11]. View-GCN [13] therefore proposes a view-based graph convolution network to capture structural relations among data, leading to sota results in multi-view image classification. Existing multi-view methods exploit mainly image data with the intention to encode the 3D shape feature by aggregating multi-view visual features [15]. Instead, our work, to the best of our knowledge, is the first work that directly processes multi-view partial PCDs for shape classification.

[1]`https://github.com/SMohammadi89/PointView-GCN`

3. METHOD

PointView-GCN aggregates multi-view shape features from partial PCDs that are captured at different view points around the object in order to improve the recognition accuracy. For each captured single-view PCD, we extract the shape feature with backbones from PCD-based sota methods for shape recognition [2, 5]. Then we construct a view graph $G = \{v_i\}_{i \in N}$ of N nodes, where each node v_i is represented by the shape feature F_i corresponding to each single-view PCD. Let $\mathbf{F} = \{F_i\}_{i \in N}$ be the features of all nodes in G. Node v_p is considered as a neighbour of node v_i if v_p is within the k-nearest neighbours (KNN) defined by its view position. The binary adjacency matrix $\mathbf{A}$ defines the neighbourhood of G.

The proposed network for feature aggregation consists of multiple levels of GCNs, as shown in Fig. 2. The number of levels M is optimally chosen according to extensive experimental analysis (see section 4.2). At each level j, we perform graph convolution on the input graph G^j to update the node feature F_i^j followed by a selective view sampling to form a smaller graph G^{j+1} with the most discriminative views selected from G^j [13] . The updated features in the new graph G^{j+1} is then fed as input to the next level of convolution. We will describe in details the graph convolution and the selective view-sampling, as well as how we fuse the global shape feature F^j at each level for the shape classification task.

3.1. Graph convolution and Selective View Sampling

With the input view-graph G^j at j^{th} level constructed with all features $\mathbf{F^j}$, we perform three main steps: i) local graph convolution and ii) non-local message passing, and 3) selective view sampling (SVS).

Firstly, the local graph convolution updates the feature of each node v_i^j considering its neighbouring nodes as $\mathbf{F}^j = \mathcal{L}\left(\mathbf{A}^j\mathbf{F}^j\mathbf{W}^j; \alpha^j\right)$, where $\mathcal{L}(\cdot)$ represents the LeakyReLU operation with related parameters α^j and $\mathbf{W}^j$ is the weight matrix. We further update features $\mathbf{F}^j$ considering the long-range relationships among all nodes in the view-graph G^j with non-local message passing. Each node v_i first updates the state of its edge with any neighbouring node v_j as a message $m_{i,p}^j = \mathcal{R}\left(F_i^j, F_p^j; \beta^j\right)_{i,p \in N^j}$, where $\mathcal{R}(\cdot)$ is the relation function among a pair of views with the related parameters β^j. We then update the feature of node v_i using itself and all the received pair-wise messages as $F_i^j = \mathcal{C}\left(F_i^j, \sum_{p=1,p\neq i}^{N_j} m_{i,p}^j; \gamma^j\right)$, where $\mathcal{C}(\cdot)$ is a combination function with related parameters γ^j. After non-local message passing, feature F_i^j is finally updated considering the relationship over the entire graph.

Furthermore, we apply SVS to select the most descriptive views among each neighbourhood. We first sub-sample the view-graph G^j using Farthest Point Sampling (FPS). Among the KNN, $\mathbf{V}_i^j$, of each sub-sampled node v_i, we then apply a view-selector that essentially selects the node with the highest response of the softmax function. Finally, the coarsened view-graph G^{j+1} with its associated updated features $\mathbf{F}^{j+1}$, is fed to the next level of graph convolution with selective view-sampling, until the last level of graph convolution.

3.2. Multi-level feature aggregation and training loss

After each local graph convolution, a max-pooling is performed on the updated features $\mathbf{F}^j$ to obtain a global shape feature F_{global}^j at each level. The final global shape feature F_{global} is the concatenation of the pooled features at all levels. We also add one residual connection from the output of the first convolution level to the last one to prevent from vanishing gradients when increasing the number of GCN levels (see Section 4.2 for detailed experimental studies). The training loss consists of two elements, i.e. global shape loss L_{global}, and the selective-view shape loss $L_{selective}$:

$$\begin{aligned} L = & L_{global}\left(\mathcal{S}(F_{global}), y\right) + \\ & \sum_{j=1}^{M}\sum_{i=1}^{N^{j+1}}\sum_{v_s \in \mathbf{V}_i^j} L_{selective}(\mathcal{V}(F_s^j; \theta^j), y), \end{aligned} \tag{1}$$

where L_{global} is the cross-entropy loss, $\mathcal{S}$ is a classifier with fully connected layer and softmax function, and y is the shape category. $L_{selective}$ is the cross-entropy loss for view selector, necessary to evaluate if the selected view can recognise the shape category. $\mathcal{V}(\cdot)$ is the function for the view selector with parameters θ^j, and F_s^j is feature of the sub-sampled node. Only L_{global} is involved at test time.

4. EXPERIMENTS

We perform extensive experiments to justify the design choices of *PointView-GCN* and to compare *PointView-GCN* against the sota 3D shape recognition methods, with our single-view datasets generated from benchmark datasets.

Table 1: Comparison with sota methods on ModelNet40. Best results with PCD inputs are highlighted in bold. Results in italic are image-based methods.

Method	Input	Views	Overall-Accuracy
MVCNN [14]	*image*	80	*90.1*
View-GCN [13]	*image*	20	*97.5*
VoxNet [17]	volume	-	85.9
Subvolume [18]	volume	-	89.2
PointNet [1]	PCD	-	89.2
PointNet++ [2]	PCD	-	90.7
DGCNN[5]	PCD	-	92.9
DGCNN	single-view PCD	1	89.0
PointNet++	single-view PCD	1	90.0
PointView-GCN(DGCNN)	single-view PCD	20	94.1
PointView-GCN(PointNet++)	single-view PCD	20	**95.4**

Dataset generation. We make use of both synthetic and real-world datasets that are publicly available for the analysis of 3D object classification. ModelNet40 [9] contains 12,311 noise-free shape models from 40 categories, while ScanObjectNN [10] contains 2909 scans of objects extracted from real-world scenes covering 15 categories. We build four single-view PCD datasets accounting self-occlusions on top of the two datasets: **Model-D**, **Model-H**, **Scan-D** and **Scan-H**, on top of ModelNet40 and ScanObjectNN under two view configurations, namely a dodecahedral (short for D) and hemispherical configurations (short for H) using Open3D rendering tools [16]. For the dataset generation, we position at the origin of the 3D mesh of each object whose dimension is normalised to $[-1, 1]$. Cameras are placed around the object and they observe parts of the object. The dodecahedral configuration has 20 viewpoints distributed on a dodecahedron surface as in [13]. Moreover, considering that viewpoints located on the lower half of the dodecahedron surface are uncommon in practice, we adopt the hemispherical configuration with only 12 views are lying on the upper part of dodecahedron surface. See supplementary material.

Implementation details. Each single-view PCD is sampled to 1024 points with FPS, and fed to the a PCD-based backbone fine-tuned on our single-view datasets, to extract the 512-d feature vector. PointNet++ [2] is used as the default feature extractor, while experiments with DGCNN [5] are also performed for comparison. To train *PointView-GCN*, we use Adam optimiser with weight decay 0.09, and adopt a learning rate warm-up strategy as in [13]. The trainable parameters are $\mathbf{W}^j$,α^j,β^j,γ^j,θ^j. Our model is trained for 20 epochs and 50 epochs on the single-view ModelNet40 and single-view ScanObjectNN, respectively.

4.1. Comparison against state-of-the-art methods

We compare *PointView-GCN* against a range of sota methods for shape classification with 3D data, e.g. volume, PCD ,and multi-view images, which are reported on benchmark ModelNet40 and the real-world ScanObjectNN datasets. More-

Table 2: Comparison with PCD-based sota methods on ScanObjectNN. Best results are highlighted in bold.

Method	Input	Views	Overall-Accuracy
SpiderCNN [19]	PCD	-	79.5
PointNet [1]	PCD	-	79.2
PointNet++ [2]	PCD	-	84.3
PointCNN [4]	PCD	-	85.5
DGCNN [5]	PCD	-	**86.2**
DGCNN	single-view PCD	1	78.4
PointNet++	single-view PCD	1	79.5
PointView-GCN(DGCNN)	single-view PCD	20	83.5
PointView-GCN(PointNet++)	single-view PCD	20	85.5

Table 3: Overall classification accuracy of *PointView-GCN* with varying number of GCN levels on the four single-view datasets. Results of *PointView-GCN* with more than 4 levels under the hemispherical configuration are not available (N.A.) due to the insufficient number of nodes at the last GCN layer.

Dataset	# GCN levels								
	1	2		3		4		5	
	no skip	no skip	skip	no skip	skip	no skip	skip	no skip	skip
Model-D	93.2	94.0	94.2	95.1	95.2	95.3	**95.4**	95.0	95.1
Model-H	93.0	93.9	94.3	94.3	94.3	94.4	**94.5**	NA	NA
Scan-D	80.8	82.9	83.0	83.2	83.7	84.4	**85.5**	83.5	83.7
Scan-H	80.3	81.2	81.3	81.4	81.5	82.7	**83.4**	NA	NA

Table 4: Overall classification accuracy of *PointView-GCN* with different number of input views on our four single-view datasets. Results of *PointView-GCN* with more than 12 input views are N.A. on Model-H and Scan-H due to the limited views under the hemispherical configuration.

Dataset	Number of input views			
	20	16	12	8
Model-D	**95.4**	95.0	94.9	92.9
Model-H	N.A.	N.A.	94.5	92.1
Scan-D	**85.5**	84.3	83.8	83.1
Scan-H	N.A.	N.A.	83.4	82.9

over, two different backbones are used as feature extractors to justify the consistent strength of *PointView-GCN* in terms of feature aggregation. Table 1, reports the results evaluated on ModelNet40. *PointView-GCN* outperforms the best-performing sota method with 3D data, i.e. DGCNN [5] by $\sim 2\%$, while approaching View-GCN [13], which has access to multi-view image data with more fine-level details of the object. Table 2 shows the results on ScanObjectNN. In general existing PCD-based methods perform worse on the real-world self-occluded PCD compared to the synthetic dataset ModelNet40. With *PointView-GCN*, we can observe the consistent improvement in performance by about $\sim 6\%$ comparing to models fine-tuned with single-view PCD, while reaching comparable performance to the best-performing method, DGCNN [5], with a margin less than 1%.

4.2. Ablation studies

We perform extensive ablation studies with our dataset generated on top of both ModelNet40 and ScanObjectNN, to justify our network design choices.

Table 5: Classification accuracy of *PointView-GCN* when fed with PointNet++ models at varying accuracy.

Dataset	Method	Overall-Accuracy			
Model-D	PointNet++	63.0	72.0	85.0	90.0
	PointView-GCN	79.0	83.0	91.0	**95.4**
Scan-D	PointNet++	61.0	74.2	78.0	79.5
	PointView-GCN	75.1	81.6	83.3	**85.5**

Effects of levels of GCN and skip connection. We evaluate how the number of GCN levels affects the classification accuracy with the four single-view datasets as well as the addition of the skip connection between the first and last layer with respect to our baseline [13], (3 GCN levels no skip connection). As shown in Table 3, with an increasing number of GCN levels, the classification accuracy increases and saturates at four levels on all tested datasets. We also notice that the skip connection helps to improve the classification accuracy by a slight margin on all datasets.

Effects of number of input views. We also evaluate how the number of input views impacts *PointView-GCN* classification performance. Table 4, reports results of our method on the four datasets with varying number of input views gradually decreasing from 20 to 8. It can be observed that there is a marginal reduction in accuracy when we reduce the number of input views from 20 to 12. However, the accuracy drops significantly when the number of views are reduced to 8. In general, we notice that with the hemispherical view configuration, the achieved classification accuracy is often slightly lower than the dodecahedron view setting. This is because the hemispherical view does not capture the bottom part of the object, yet it reflects the real-world capturing [20, 13].

Effects of PointNet++ models of varying classification accuracy. Finally, we use the extracted feature from a set of pre-trained PointNet++ models achieving different classification accuracy. Table 5, shows the consistent capability of *PointView-GCN* for feature aggregation. With the weakest single-view shape feature, *PointView-GCN* increases the classification accuracy by $\sim 15\%$, while for the strongest single-view shape feature, there is still $\sim 5\%$ improvement.

5. CONCLUSION

We proposed *PointView-GCN* for shape classification by aggregating multi-view shape features with a multi-level GCN. With our novel single-view datasets generated on top of benchmark datasets, we proved that *PointView-GCN* is able to produce a more discriminative shape feature that outperforms the sota PCD-based methods on the benchmark dataset. As future work, we will further investigate the fusion of multi-view visual and shape features within the framework of *PointView-GCN*, to boost the 3D classification accuracy.

6. REFERENCES

[1] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas 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.

[2] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, 2017, pp. 5099–5108.

[3] Anastasia Ioannidou, Elisavet Chatzilari, Spiros Nikolopoulos, and Ioannis Kompatsiaris, "Deep learning advances in computer vision with 3d data: A survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–38, 2017.

[4] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen, "Pointcnn: Convolution on x-transformed points," *Advances in neural information processing systems*, vol. 31, pp. 820–830, 2018.

[5] Anh Viet Phan, Minh Le Nguyen, Yen Lam Hoang Nguyen, and Lam Thu Bui, "Dgcnn: A convolutional neural network over large-scale labeled graphs," *Neural Networks*, vol. 108, pp. 533–543, 2018.

[6] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari, "3d point capsule networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 1009–1018.

[7] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 918–927.

[8] Yiming Wang, Marco Carletti, Francesco Setti, Marco Cristani, and Alessio Del Bue, "Active 3d classification of multiple objects in cluttered scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 2602–2610.

[9] Kashi Venkatesh Vishwanath, Diwaker Gupta, Amin Vahdat, and Ken Yocum, "Modelnet: Towards a datacenter emulation environment," in *Proceedings of the IEEE Ninth International Conference on Peer-to-Peer Computing*. IEEE, 2009, pp. 81–82.

[10] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," *arXiv*, pp. arXiv–1908, 2019.

[11] Shallu Sharma and Rajesh Mehra, "Implications of pooling strategies in convolutional neural networks: A deep insight," *Foundations of Computing and Decision Sciences*, vol. 44, no. 3, pp. 303–330, 2019.

[12] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, pp. 11, 2019.

[13] Xin Wei, Ruixuan Yu, and Jian Sun, "View-gcn: View-based graph convolutional network for 3d shape analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1850–1859.

[14] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *in Proceedings of the IEEE international conference on computer vision*, 2015, pp. 945–953.

[15] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan, "A survey on deep learning-based fine-grained object classification and semantic segmentation," *International Journal of Automation and Computing*, vol. 14, no. 2, pp. 119–135, 2017.

[16] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun, "Open3d: A modern library for 3d data processing," *arXiv preprint arXiv:1801.09847*, 2018.

[17] Daniel Maturana and Sebastian Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 922–928.

[18] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas, "Volumetric and multi-view cnns for object classification on 3d data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5648–5656.

[19] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao, "Spidercnn: Deep learning on point sets with parameterized convolutional filters," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 87–102.

[20] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida, "Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5010–5019.