Points of Comparison: A Study of 3D Point Cloud Networks for Robustness Against Data Corruptions

Charchit Saraswat Debanjan Mondal
Harsha Kanaka Eswar Gudipudi
Manning College of Information and Computer Sciences
University of Massachusetts Amherst

Abstract

With the increasing popularity of 3D Point Cloud based Neural Network techniques in safety-critical techniques, it has become increasingly important to benchmark their performance against datasets that closely represent real-world artifacts. Real-world data collection suffers from irregularities like noise, occlusion, and transformations. A dataset that closely simulates such conditions through corruption is ModelNet40-C. We benchmark performance of the Point Transformer model on the ModelNet40-C dataset and show that data augmentations improve the performance of the model from baseline ModelNet40 classification tasks. To the best of our knowledge, our work is the first effort to integrate the Point Transformer with ModelNet40-C evaluation benchmark.

1. Introduction

Point clouds have gained significant recognition as the most popular and accessible data format within the realm of 3D computer vision tasks. They can be obtained from a diverse set of sensors and computer-aided design (CAD) models and are also quite flexible as representations. Due to these inherent advantages, point clouds have witnessed a growing utilization in practical applications. Recently, the usage of point clouds has gained traction in safety-critical areas like self-driving cars [17], robotics [7], medical imaging [13], and virtual and augmented reality [6].

The development of point cloud-based neural classification techniques has significantly advanced the field of 3D perception and understanding. Three prominent methods in this domain are PointNet [8], PointNet++ [9], and KPConv [11]. PointNet was one of the pioneering approaches that directly processed unordered point cloud data without requiring any additional pre-processing steps and employed shared multi-layer perceptrons. Building upon PointNet, PointNet++ introduced a hierarchical neural network architecture that captures local and global structures. KpConv, on the other hand, introduced a novel convolutional operator designed specifically for irregularly sampled point clouds. It used adaptive kernel point sampling and weighted kernel convolution.

Recently, inspired by the success of transformer-based networks in Natural Language Processing tasks[12], researchers have developed transformer-based point cloud networks which have achieved state-of-the-art results in various 3D point cloud recognition tasks [19, 15]. However, since these models will potentially be used in safety-critical applications like autonomous driving, enhancing the resilience of these models to a broad spectrum of corruptions is crucial and requires a comprehensive analysis and improvement of their robustness.

These applications require real-time predictions and hence techniques that remove noise in real-time, do completions of missing points in point-clouds, or surface parameterization in real time due to camera angle will slow the predictions and can be hazardous. Models need to be pre-trained to handle these real-world artifacts, and benchmarking these models against such data is crucial before their integration into these applications.

Given the real-world constraints, it is crucial to acknowledge that sensor inaccuracies and physical constraints introduce common corruptions in point cloud data. For instance, occlusion, a prevalent corruption in LiDAR and other scanning devices, leads to partially visible point clouds. Additionally, deformation is widespread in AR/VR games. These corruptions pose an even greater risk in real-world applications. Consequently, there is a pressing need to explore the robustness of 3D point cloud recognition against such corruption. In this work, we evaluate Point Transformer [19] on the 3D point cloud classification robustness benchmark Modelnet40-C [10].

2. Related Work

PointNet[8] emerged as one of the pioneering deep learning methodologies for point cloud recognition, employing stacked MLP layers to capture the global structure. Building upon this, PointNet++[9] extended the approach to incorporate considerations for both local and global structures.

While convolution-based approaches were originally developed for 2D image processing, they have also demonstrated noteworthy performance in point cloud analysis. Notably, KPConv[11] introduced a flexible and efficient convolution method that utilizes kernel points to determine convolution weights in Euclidean space. Moreover, KPConv can be extended to deformable convolutions, enabling adaptation of kernel points to local geometry and ensuring robustness to varying densities. Additionally, graph-based models such as the one proposed by [14] have leveraged message passing algorithms, showcasing significant performance gains.

Inspired by self-attention theory [12] in NLP, ViT [2] has shown remarkable success in 2D image understanding tasks. Since 3D point clouds are inherently unordered, utilizing self-attention appears to be a more intuitive approach. PCT[3] was the first attempt to apply the transformer architecture in the Point Cloud Recognition tasks. Later, Point Transformer [19] utilized the attetion mechanism to achieve SOTA results in 3D point cloud understanding tasks. Instead of global attention used in earlier works like PCT [3], they applied self-attention around local neighborhoods. Also, they introduced the idea of vector self-attention which captures more complexity compared to the traditional scalar dot product attention. They were able to achieve SOTA results in classification, instance, and semantic segmentation.

There is great emphasis on increasing the importance of incorporating powerful 3D shape representations with the availability of affordable 2.5-depth sensors like Microsoft Kinect. ModelNet40 [16] introduces a Convolutional Deep Belief Network-based approach called 3DShapeNets, which represents complex 3D shapes as probability distributions of binary variables on a 3D voxel grid. The proposed model learns to shape distributions from CAD data, discovers hierarchical part representations automatically, and supports joint object recognition and shape completion from 2.5D depth maps. The authors have constructed a large-scale 3D CAD model dataset called ModelNet for training the 3D deep learning model. Extensive experiments demonstrate the superiority of their 3D deep representation over existing methods across various tasks. The dataset contains shapes from 40 categories, which are split into training and testing. ModelNet40-C [10], a benchmark for evaluating corruption robustness in 3D point cloud models. They identify a significant performance gap between state-of-the-art models on ModelNet40 and ModelNet40-C. To address this gap, the authors propose a simple yet effective method by combining PointCutMix-R and TENT. They emphasize the strength of Transformer-based architectures with proper training recipes for achieving robustness.

3. Method and Architecture

3.1. Dataset

ModelNet40 dataset consists of 40 different object categories, encompassing common objects such as chairs, tables, lamps, cars, airplanes, and more. The original ModelNet40 consists of 12,311 CAD-generated meshes, out of which 9,843 are used for training while the rest 2,468 are reserved for testing. The corresponding point cloud data points are uniformly sampled from the mesh surfaces, and then further preprocessed by moving to the origin and scaling into a unit sphere.

ModelNet40-C is a systemic corruption robustness benchmark based on the ModelNet40 dataset. The dataset encompasses different common corruptions observed in sensor, LIDARs and AR/VR systems. Overall, it has 15 corruption types, each with 5 severity levels. This makes it a 75x larger dataset than the original ModelNet40. Note that, ModelNet40-C exclusively utilizes the test split of the clean dataset. Therefore, it's not meant to be used during the training phase, but rather intended for evaluating the robustness of point-cloud based models. The dataset contains 185,000 distinct point clouds.

The 15 corruption types can be divided into 3 broad categories. The three broad categories are density, noise and transformations. Each of these have 5 sub-categories. For density, they are occlusion, lidar, local_density_inc, local_density_dec and cutout. Occlusion and LiDAR techniques employ ray tracing on original meshes to simulate occlusion patterns, with LiDAR further incorporating the vertical line-styled pattern of LiDAR point clouds. Additionally, the methods of Local Density Inc, Local Density Dec, and Cutout leverage k nearest neighbors (kNN) to randomly select and modify local clusters of points to either increase or decrease their density. For noise, different noises applied are uniform, gaussian, impulse, upsampling and background. Uniform and Gaussian methods introduce distinct distributional noise to individual points in a point cloud. Impulse applies deterministic perturbations to a subset of points, while upsampling generates new perturbation points around existing points. Additionally, the background technique randomly adds new points within the bounding box space of the original point cloud. Transformations like rotation, shear, FFD, RBF and INV_RBF are applied to the dataset. Real-world point clouds often undergo rotations, and the robustness against adversarial rotations has been explored in various studies. In this context, the implementation considers mild rotations in xyz plane, Shear on the

xy plane as a representative motion distortion in 3D point clouds and investigates the application of Free-form deformation (FFD) and radial basis function (RBF)-based deformation for non-linear transformations. These corruptions are represented in Fig1.

3.2. Training

The ModelNet40-C paper conducted their analysis on 6 different archetectures, namely PointNet [8], PointNet++ [9], DGCNN [14], RSCNN [5], PCT [3]. We evaluated a new architecture, Point Transformer on this benchmark. It was introduced after the ModelNet40-C paper was published. Point Transformer is based on the Transformer architecture, and utilizes vector self-attention in the transformer block. It has achieved remarkable results on ModelNet40 classification and segmentation benchmarks.

We conducted our training experiments on NVIDIA Tesla M40 GPU. The original paper recommended running 200 epochs. However, we ran the model for 75 epochs only due to resource and time constraints. Our entire training procedure 100 hours and testing took an additional 30 hours.

3.2.1 Training Data Augmentation

We explored four types of training data augmentation that were mentioned in the ModelNet40-C paper.

- **PointCutMix:[18]** PointCutMix optimally matches points between two point clouds, generating new training data by replacing points with their optimal assigned pairs. It employs two replacement strategies: random selection of all replacement points or selecting k nearest neighbors of a random point. Hence PointCutMix-R and PointCutMix-K. These strategies consistently enhance performance in point cloud classification tasks. The introduction of saliency maps for point selection further improves performance and enhances model robustness against point attacks.
- **PointMixup[1]:** PointMixup is a data augmentation method for point clouds, leveraging interpolation techniques from the image domain. It addresses the lack of one-to-one correspondence between points in different objects by employing a shortest path linear interpolation approach. By optimally assigning a path function, PointMixup generates new examples that follow the shortest path and allows for the application of interpolation-based regularizers such as mixup and manifold mixup to improve regularization in the point cloud domain.
- **RSMix[4]:** RSMix generates virtual mixed samples by replacing a portion of one sample with shape-preserved subsets from another sample, preserving the structural integrity of the point cloud. The neighboring function in RSMix is designed to handle the unordered and non-grid nature of point clouds, ensuring the preservation of their unique properties during augmentation.

3.2.2 Training and Evaluation Procedure

We trained 5 instances of the Point Transformer. For the first one, we trained using the standard training procedure mentioned in this implementation[1] by one of the authors. For the other four instances, we utilized PointCutMix-K, PointCutMix-R, PointMixup and RSMix data augmentations respectively. These augmentations were applied during training runtime to reduce storage space. We reused the data augmentation codes from the ModelNet40-C implementation[2]. We chose the hyperparameters, optimizer and loss based on the original implementation. However, due to resource constarints, we ran the training for 75 epochs compared to 200 epochs mentioned in the original implementation. We employed cross-entropy loss, Adam optimizer and step scheduler during our training.

After training, we tested these 5 models on the ModelNet40 test split and ModelNet40-C dataset. For ModelNet40-C, we conducted the evaluation separately for all models, corruption types and severity levels. Hence, we ran $75 \times 5 = 375$ different evaluation runs and aggregated the results.

4. Experiments

4.1. Clean Data

Initially we measured the effects of training data augmentations on the clean ModelNet40 data. The evaluation results on the clean data is shown in Table 1. The original accuracy reported in the Point Transformer paper on ModelNet40 test split was 93.7%. We obtained test accuracy of 88.89% with original training method, which maybe attributed to training less no of epochs. We also observe that all data augmentation strategies outperform the model with no augmentations(refered to as original in all the tables) in terms of accuracy and mean class accuracies. This unequivocally demonstrates the substantial utility of all the aforementioned data augmentations in Point Cloud Recognition tasks. RSMix performs the best in terms of accuracy, whereas PointCutMix-K displays the best mean class accuracy.

4.2. ModelNet40-C Corrupted Data

As discussed earlier, we conducted a total of 375 experiments. We evaluated our 5 models on the 15 corruption

[1]https://github.com/qq456cvb/Point-Transformers.git
[2]https://github.com/jiachens/ModelNet40-C

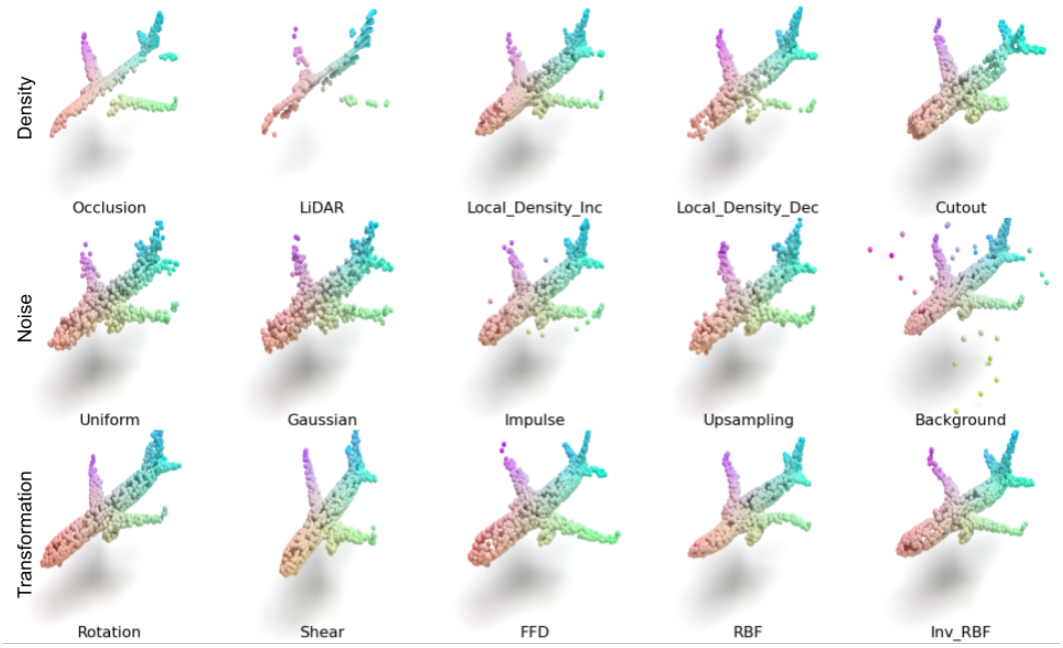

Figure 1: The corruptions applied to ModelNet40 dataset to create ModelNet40-C.

Augmentation	Accuracy	Mean Class Accuracy
Original	0.8889	0.8524
PointCutMix-K	0.8962	**0.8634**
PointCutMix-R	0.8954	0.8598
PointMixup	0.8995	0.8622
RSMix	**0.9039**	0.8621

Table 1: Summary of PTv1 mean model and class accuracies with and without augmentation.

types available in ModelNet40-C, each having 5 severity levels. After obtaining those results, we aggregate across the severity levels to get the accuracy and mean class accuracy scores for each corruption type. These detailed results are presented in 3. In order to gain a more comprehensive and broader understanding, we further aggregate across similar corruption types to obtain a score for each corruption category. We report the tabulated results in Table 2.

We observed the following trends after analyzing the different corruptions.

- As visualized in Figure and tabulated in Table 2, we observe that all augmentation strategies significantly outperform the baseline no augmentation strategy by a significant margin. The disparity in performance is particularly pronounced in the case of Noise corruptions, with the superior strategy RSMix exhibiting a 26% higher accuracy compared to the baseline approach. For density and transformation corruptions, the performance gap is 14% and 18% respectively. Even the worst performing augmentation strategy shows a notable improvement over to the baseline strategy. This highlights the significance of employing proficient augmentation techniques in order to achieve improved performance on real-world corrupted data.

- As observed in Table 3, RSMix demonstrates superior performance across all corruption types, with the exception of Background corruption. Notably, in the original ModelNet40-C paper, RSMix exhibited the highest performance for PCT (Point Cloud Transformer), another transformer-based model. Therefore, we can ascertain that RSMix emerges as the optimal augmentation strategy for transformer-based point-cloud models.

- One significant discovery highlighted in the ModelNet40-C paper was the resilience of transformer-based architectures, such as PCT, against transformation corruptions. The results depicted in Figure 2 and Table 2 distinctly illustrate that Point Transformer outperforms in transformation corruptions in comparison to density and noise corruptions. This reaffirms the assertion that transformer-based architectures are indeed robust in the face of transformation corruptions.

- Table 3 reveals a noteworthy discrepancy, wherein the baseline model attains a mere 6% accuracy on

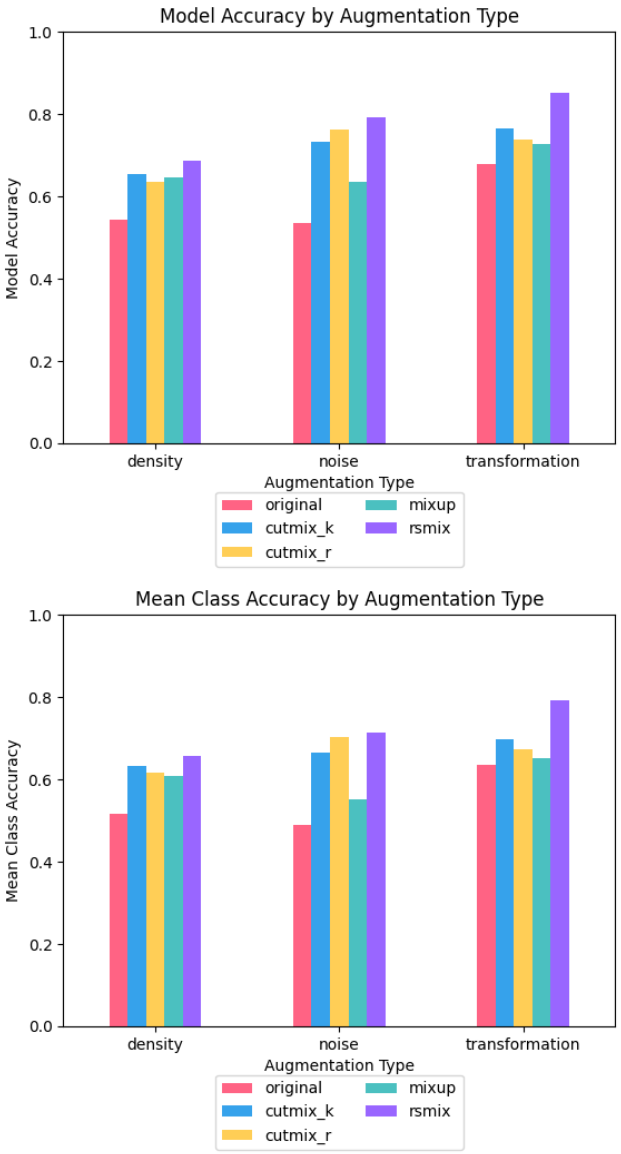

Figure 2: (a) Accuracy of PTv1 with 4 augmentations and on original ModelNet40-C (b) Mean Class Accuracy of PTv1 with 4 augmentations and on original ModelNet40-C

the background corruption, while the top-performing PointCutMix-R achieves a commendable accuracy of 69%. This stark contrast demonstrates that models trained solely on clean data exhibit considerable vulnerability to background noise, underscoring the critical necessity of employing data augmentation techniques when addressing background noises in Point Clouds.

Corruption Category	Augmentation	Accuracy	Mean Class Accuracy
density	Original	0.5425	0.5169
	PointCutMix-K	0.6550	0.6314
	PointCutMix-R	0.6359	0.6173
	PointMixup	0.6454	0.6091
	RSMix	**0.6857**	**0.6582**
noise	Original	0.5354	0.4891
	PointCutMix-K	0.7319	0.6647
	PointCutMix-R	0.7622	0.7030
	PointMixup	0.6361	0.5520
	RSMix	**0.7912**	**0.7148**
transformation	Original	0.6775	0.6365
	PointCutMix-K	0.7655	0.6984
	PointCutMix-R	0.7377	0.6730
	PointMixup	0.7265	0.6502
	RSMix	**0.8506**	**0.7921**

Table 2: Summary of PTv1 mean model and mean class accuracies on the 3 data corruptions types with and without augmentation during training.

5. Conclusion

In conclusion, considering most existing point cloud datasets consists of clean denoised data, it becomes imperative to supplement model training with augmented data, especially for safety-critical applications. This approach enhances the robustness of models against prevalent data corruptions, as evidenced by benchmarking experiments conducted on ModelNet40-C. Our study convincingly demonstrates that incorporating data augmentation strategies significantly improves performance on corrupted datasets. thereby reinforcing the importance of this technique in practical applications.

References

[1] Yunlu Chen, Vincent Tao Hu, Efstratios Gavves, Thomas Mensink, Pascal Mettes, Pengwan Yang, and Cees G. M. Snoek. Pointmixup: Augmentation for point clouds, 2020. 3

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 2

[3] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. PCT: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, apr 2021. 2, 3

[4] Dogyoon Lee, Jaeha Lee, Junhyeop Lee, Hyeongmin Lee, Minhyeok Lee, Sungmin Woo, and Sangyoun Lee. Regularization strategy for point cloud via rigidly mixed sample, 2021. 3

[5] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis, 2019. 3

[6] Peter M Maloca, J. Emanuel Ramos de Carvalho, Tjebo Heeren, Pascal W Hasler, Faisal Mushtaq, Mark Mon-

	Density					Noise					Transformation				
	Occlusion	**LIDAR**	**Local_Density_Inc**	**Local_Density_Dec**	**Cutout**	**Uniform**	**Gaussian**	**Impulse**	**Upsampling**	**Background**	**Rotation**	**Shear**	**FFD**	**RBF**	**Inv_RBF**
Original	0.2685	0.1766	0.8033	0.7068	0.7573	0.6586	0.8148	0.5233	0.618	0.0622	0.6481	0.6631	0.6863	0.6951	0.6951
PointCutMix-K	0.3673	0.2696	0.8837	0.8706	0.8839	0.7317	0.8743	0.6981	0.7125	0.6429	0.7497	0.7568	0.7656	0.7785	0.7767
PointCutMix-R	0.3387	0.2351	0.8763	0.8622	0.8672	0.7509	0.8806	0.781	0.7019	**0.6963**	0.7036	0.7218	0.7461	0.7571	0.7601
PointMixup	0.3716	0.2777	0.876	0.8421	0.8596	0.6955	0.8759	0.6403	0.7117	0.2572	0.6935	0.7142	0.7357	0.7423	0.7465
RSMix	**0.439**	**0.3108**	**0.8959**	**0.8918**	**0.891**	**0.838**	**0.889**	**0.8041**	**0.8224**	0.6026	**0.8188**	**0.8539**	**0.8553**	**0.8608**	**0.8642**

Table 3: Tabular Summary of PTv1 accuracies on the 15 data corruptions with and without augmentation during training.

Williams, Hendrik P.N. Scholl, Konstantinos Balaskas, Catherine Egan, Adnan Tufail, Lilian Witthauer, and Philippe C. Cattin. High-Performance Virtual Reality Volume Rendering of Original Optical Coherence Tomography Point-Cloud Data Enhanced With Real-Time Ray Casting. *Translational Vision Science Technology*, 7(4):2–2, 07 2018. 1

[7] François Pomerleau, Francis Colas, and Roland Siegwart. 2015. 1

[8] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017. 1, 2, 3

[9] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, 2017. 1, 2, 3

[10] Jiachen Sun, Qingzhao Zhang, Bhavya Kailkhura, Zhiding Yu, Chaowei Xiao, and Z. Morley Mao. Benchmarking robustness of 3d point cloud recognition against common corruptions, 2022. 1, 2

[11] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds, 2019. 1, 2

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. 1, 2

[13] Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration, 2019. 1

[14] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds, 2019. 2, 3

[15] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling, 2022. 1

[16] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes, 2015. 2

[17] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking, 2021. 1

[18] Jinlai Zhang, Lyujie Chen, Bo Ouyang, Binbin Liu, Jihong Zhu, Yujing Chen, Yanmei Meng, and Danfeng Wu. Pointcutmix: Regularization strategy for point cloud classification, 2021. 3

[19] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer, 2021. 1, 2