CP^3: Channel Pruning Plug-in for Point-based Networks

Yaomin Huang[1,2,*] Ning Liu[2,*] Zhengping Che[2] Zhiyuan Xu[2] Chaomin Shen[1]
Yaxin Peng[3] Guixu Zhang[1,✉] Xinmei Liu[1] Feifei Feng[2] Jian Tang[2,✉]
[1]School of Computer Science, East China Normal University
[2]Midea Group
[3]Department of Mathematics, School of Science, Shanghai University
[1]{51205901049,51205901078}@stu.ecnu.edu.cn [1]{cmshen,gxzhang}@cs.ecnu.edu.cn
[2]{liuning22,chezp,xuzy70,feifei.feng,tangjian22}@midea.com [3]yaxin.peng@shu.edu.cn

Abstract

Channel pruning can effectively reduce both computational cost and memory footprint of the original network while keeping a comparable accuracy performance. Though great success has been achieved in channel pruning for 2D image-based convolutional networks (CNNs), existing works seldom extend the channel pruning methods to 3D point-based neural networks (PNNs). Directly implementing the 2D CNN channel pruning methods to PNNs undermine the performance of PNNs because of the different representations of 2D images and 3D point clouds as well as the network architecture disparity. In this paper, we proposed CP^3, which is a ***C****hannel* ***P****runing* ***P****lug-in for* ***P****oint-based network. CP^3 is elaborately designed to leverage the characteristics of point clouds and PNNs in order to enable 2D channel pruning methods for PNNs. Specifically, it presents a coordinate-enhanced channel importance metric to reflect the correlation between dimensional information and individual channel features, and it recycles the discarded points in PNN's sampling process and reconsiders their potentially-exclusive information to enhance the robustness of channel pruning. Experiments on various PNN architectures show that CP^3 constantly improves state-of-the-art 2D CNN pruning approaches on different point cloud tasks. For instance, our compressed PointNeXt-S on ScanObjectNN achieves an accuracy of 88.52% with a pruning rate of 57.8%, outperforming the baseline pruning methods with an accuracy gain of 1.94%.*

1. Introduction

Convolutional Neural Networks (CNNs) often encounter the problems of overloaded computation and overweighted storage. The cumbersome instantiation of a CNN model leads to inefficient, uneconomic, or even impossible deployment in practice. Therefore, light-weight models that provide comparable results with much fewer computational costs are in great demand for nearly all applications. Channel pruning is a promising solution to delivering efficient networks. In recent years, 2D CNN channel pruning, e.g., pruning classical VGGNets [37], ResNets [14], MobileNets [16], and many other neural networks for processing 2D images [6, 7, 12, 24, 26, 29, 40], has been successfully conducted. Most channel pruning approaches focus on identifying redundant convolution filters (i.e., channels) by evaluating their importance. The cornerstone of 2D channel pruning methods is the diversified yet effective channel evaluation metrics. For instance, HRank [24] uses the rank of the feature map as the pruning metric and removes the low-rank filters that are considered to contain less information. CHIP [40] leverages channel independence to represent the importance of each feature mapping and eliminates less important channels.

With the widespread application of depth-sensing technology, 3D vision tasks [9, 10, 36, 44] are a rapidly growing field starving for powerful methods. Apart from straightforwardly applying 2D CNNs, models built with Point-based Neural Networks (PNNs), which directly process point clouds from the beginning without unnecessary rendering, show their merits and are widely deployed on edge devices for various applications such as robots [22, 49] and self-driving [2, 53]. Compressing PNNs is crucial due to the limited resources of edge devices and multiple models for different tasks are likely to run simultaneously [8, 30]. Given the huge success of 2D channel pruning and the great

[*] Equal contributions.
[✉] Corresponding authors.
This work is done during Yaomin Huang and Xinmei Liu's internship at Midea Group.

demand for efficient 3D PNNs, we intuitively raise one question: *shall we directly implement the existing pruning methods to PNNs following the proposed channel importance metrics in 2D CNNs pruning?*

With this question in mind, we investigate the fundamental factors that potentially impair 2D pruning effectiveness on PNNs. Previous works [19, 48] have shown that point clouds record visual and semantic information in a significantly different way from 2D images. Specifically, a point cloud consists of a set of unordered points on objects' and environments' surfaces, and each point encodes its features, such as intensity along with the spatial coordinates (x, y, z). In contrast, 2D images organize visual features in a dense and regular pixel array. Such data representation differences between 3D point clouds and 2D images lead to a) different ways of exploiting information from data and b) contrasting network architectures of PNNs and 2D CNNs. It is credible that only the pruning methods considering the two aspects (definitely not existing 2D CNN pruners) may obtain superior performance on PNNs.

From the perspective of data representations, 3D point clouds provide more 3D feature representations than 2D images, but the representations are more sensitive to network channels. To be more specific, for 2D images, all three RGB channels represent basic information in an isotropic and homogeneous way so that the latent representations extracted by CNNs applied to the images. On the other hand, point clouds explicitly encode the spatial information in three coordinate channels, which are indispensable for extracting visual and semantic information from other channels. Moreover, PNNs employ the coordinate information in multiple layers as concatenated inputs for deeper feature extraction. Nevertheless, existing CNN pruning methods are designed only suitable for the plain arrangements of 2D data but fail to consider how the informative 3D information should be extracted from point clouds.

Moreover, the network architectures of PNNs are designed substantially different from 2D CNNs. While using smaller kernels [37] is shown to benefit 2D CNNs [37], it does not apply to networks for 3D point clouds. On the contrary, PNNs leverage neighborhoods at multiple scales to obtain both robust and detailed features. The reason is that small neighborhoods (analogous to small kernels in 2D CNNs) in point clouds consist of few points for PNNs to capture robust features. Due to the necessary sampling steps, the knowledge insufficiency issue becomes more severe for deeper PNN layers. In addition, PNNs use the random input dropout procedure during training to adaptively weight patterns detected at different scales and combine multi-scale features. This procedure randomly discards a large proportion of points and loses much exclusive information of the original data. Thus, the architecture disparity between 2D CNNs and PNNs affects the performance of directly applying existing pruning methods to PNNs.

In this paper, by explicitly dealing with the two characteristics of 3D task, namely the data representation and the PNN architecture design, we propose a **C**hannel **P**runing **P**lug-in for **P**oint-based network named CP^3, which can be applied to most 2D channel pruning methods for compressing PNN models. The proposed CP^3 refines the channel importance, the key factor of pruning methods, from two aspects. Firstly, considering the point coordinates (x, y, and z) encode the spatial information and deeply affects feature extraction procedures in PNN layers, we determine the channel importance by evaluating the correlation between the feature map and its corresponding point coordinates by introducing a coordinate-enhancement module. Secondly, calculating channel importance in channel pruning is data-driven and sensitive to the input, and the intrinsic sampling steps in PNN naturally makes pruning methods unstable. To settle this problem, we make full use of the discarded points in the sampling process via a knowledge recycling module to supplement the evaluation of channel importance. This reduces the data sampling bias impact on the channel importance calculation and increases the robustness of the pruning results. Notably, both the coordinates and recycled points in CP^3 do not participate in network training (with back-propagation) but only assist channel importance calculation in the reasoning phase. Thus, CP^3 does not increase any computational cost of the pruned network. The contributions of this paper are as follows:

- We systematically consider the characteristics of PNNs and propose a channel pruning plug-in named CP^3 to enhance 2D CNN channel pruning approaches on 3D PNNs. To the best of our knowledge, CP^3 is the first method to export existing 2D pruning methods to PNNs.
- We propose a coordinate-enhanced channel importance score to guide point clouds network pruning, by evaluating the correlation between feature maps and corresponding point coordinates.
- We design a knowledge recycling pruning scheme that increases the robustness of the pruning procedure, using the discarded points to improve the channel importance evaluation.
- We show that using CP^3 is consistently superior to directly transplanting 2D pruning methods to PNNs by extensive experiments on three 3D tasks and five datasets with different PNN models and pruning baselines.

2. Related Work

2.1. 2D Channel Pruning

Channel pruning (a.k.a., filter pruning) methods reduce the redundant filters while maintaining the original structure of CNNs and is friendly to prevailing inference acceleration engines such as TensorFlow-Lite (TFLite) [11] and

Mobile Neural Network (MNN) [18]. Mainstream channel pruning methods [6, 7, 12, 29] usually first evaluate the importance of channels by certain metrics and then prune (i.e., remove) the less important channels. Early work [21] uses the l_1 norm of filters as importance score for channel pruning. Afterwards, learning parameters, such as the scaling factor γ in the batch norm layer [26] and the reconstruction error in the final network layer [51], are considered as the importance scores for channel selection. The importance sampling distribution of channels [23] is also used for pruning. Recent works [15, 40] measure the correlation of multiple feature maps to determine the importance score of the filter for pruning. HRank [24] proposes a method for pruning filters based on the theory that low-rank feature maps contain less information. [50] leverages the statistical distribution of activation gradient and takes the smaller gradient as low importance score for pruning. [46] calculates the average importance of both the input feature maps and their corresponding output feature maps to determine the overall importance. [13, 45] compress CNNs from multiple dimensions While most channel pruning methods are designed for and tested on 2D CNNs, our CP^3 can work in tandem with existing pruners for 3D point-based networks.

2.2. Point-based Networks for Point Cloud Data

Point-based Neural Networks (PNNs) directly process point cloud data with a flexible range of receptive field, have no positioning information loss, and thus keep more accurate spatial information. As a pioneer work, PointNet [32] learns the spatial encoding directly from the input point clouds and uses the characteristics of all points to obtain the global representations. PointNet++ [33] further proposes a multi-level feature extraction structure to extract local and global features more effectively. KPConv [42] proposes a new point convolution operation to learn local movements applied to kernel points. ASSANet [34] proposes a separable set abstraction module that decomposes the normal SA module in PointNet++ into two separate learning phases for channel and space. PointMLP [28] uses residual point blocks to extract local features, transforms local points using geometric affine modules, and extracts geometric features before and after the aggregation operation. PointNeXt [35] uses inverted residual bottleneck and separable multilayer perceptrons to achieve more efficient model scaling. Besides classification, PNNs also serve as backbones for other 3D tasks. VoteNet [31] effectively improves the 3D object detection accuracy through the Hough voting mechanism [20]. PointTransformer [52] designs models improving prior work across domains and tasks. GroupFree3D [27] uses the attention mechanism to automatically learn the contribution of each point to the object. In this paper, we show that CP^3 can be widely applied to point-based networks on a variety of point cloud benchmarks and representative original networks.

3. Methodology

Although point-based networks are similar to CNN in concrete realization, they have fundamental differences in data representation and network architecture design. To extend the success of CNN pruning to PNN, two modules are proposed in CP^3 taking advantage from the dimensional information and discarded points: 1) coordinate-enhancement (CE) module, which produces a coordinate-enhanced score to estimate the channel importance by combining dimensional and feature information, and 2) knowledge recycling module reusing the discarded points to improve the channel importance evaluation criteria and increase the robustness.

3.1. Formulations and Motivation

Point-based networks PNN is a unified architecture that directly takes point clouds as input. It builds hierarchical groups of points and progressively abstracts larger local regions along the hierarchy. PNN is structurally composed by a number of set abstraction (SA) blocks. Each SA block consists of 1) a sampling layer iteratively samples the farthest point to choose a subset of points from input points, 2) a group layer gathers neighbors of centroid points to a local region, 3) a set of shared Multi-Layer Perceptrons (MLPs) to extract features, and 4) a reduction layer to aggregate features in the neighbors. Formally speaking, a SA block takes an $n_{i-1} \times (d + c_{i-1})$ matrix as input that is from n_{i-1} points with d-dim coordinates and c_{i-1}-dim point feature. It outputs an $n_i \times (d + c_i)$ matrix of n_i subsampled points with d-dimensional coordinates (i.e., $d = 3$) and new c_i-dimensional feature vectors summarizing local context. The SA block is formulated as:

$$\mathbf{F}_i^{l+1} = \mathcal{R}\left\{h_\Theta\left(\left[\mathbf{F}_j^l; \mathbf{x}_j^l - \mathbf{x}_i^l\right]\right)\right\}, \tag{1}$$

where h_Θ is MLPs to extract grouped points feature, $\mathcal{R}$ is the reduction layer (e.g. max-pooling) to aggregate features in the neighbors $\{j : (i, j) \in \mathcal{N}\}$, F_j^l is the features of neighbor j in the l-th layer, x_i^l and x_j^l are input points coordinates and coordinates of neighbor j in the l-th layer.

Channel pruning Assume a pre-trained PNN model has a set of K convolutional layers, and $\mathcal{A}^l$ is the l-th convolution layer. The parameters in $\mathcal{A}^l$ can be represented as a set of filters $\mathcal{W}_{\mathcal{A}^l} = \left\{\mathbf{w}_1^l, \mathbf{w}_2^l, \ldots, \mathbf{w}_{c_l}^l\right\} \in \mathbb{R}^{(d+c_l)\times(d+c_{l-1})\times k_l \times k_l}$, where j-th filter is $\mathbf{w}_j^l \in \mathbb{R}^{(d+c_{l-1})\times k_l \times k_i}$. $(d + c_l)$ represents the number of filters in $\mathcal{A}^l$ and k_l denotes the kernel size. The outputs of filter, i.e., feature map, are denoted as $\mathcal{F}^l = \left\{\mathbf{f}_1^l, \mathbf{f}_2^l, \ldots, \mathbf{f}_{n_i}^l\right\} \in \mathbb{R}^{n_i \times (d+c_i)}$. Channel pruning aims to identify and remove the less importance filter from the

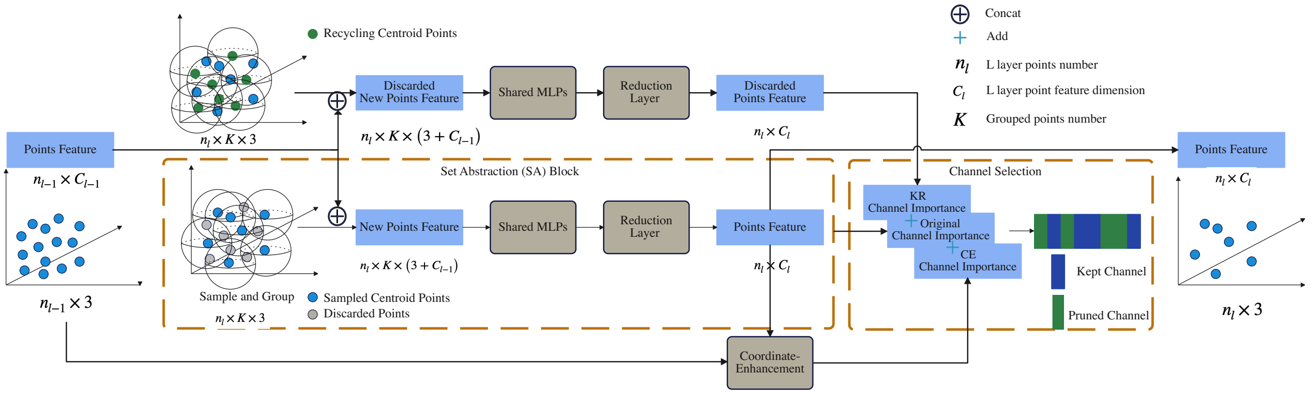

Figure 1. The framework of CP^3. The figure shows the specific pruning process of one of the SA blocks. Whether a channel in a PNN is pruned is determined by three parts: 1) Original channel importance: obtained from the original CNNs channel pruning method (e.g., HRank [24], CHIP [40]). 2) Discarded channel importance: obtained from the Knowledge-Recycling module by leveraging the discarded points in the network to supplement the channel importance evaluation of the corresponding points and improve the robustness of the channel selection. 3) CE (Coordinate-Enhanced) channel importance: obtained from calculating the correlation between the feature map and its corresponding points coordinates to guide point clouds network pruning.

original networks. In general, channel pruning can be formulated as the following optimization problem:

$$\min_{\delta_{ij}} \sum_{i=1}^{K} \sum_{j=1}^{n_i} \delta_{ij} \mathcal{L}\left(\mathbf{w}_j^i\right), \text{ s.t. } \sum_{j=1}^{n_i} \delta_{ij} = k_l, \tag{2}$$

where δ_{ij} is an indicator which is 1 if $\mathbf{w}_j^i$ is to be pruned or 0 if $\mathbf{w}_j^i$ is to be kept, $\mathcal{L}(\cdot)$ measures the importance of a filter and k_l is the kept filter number.

Robust importance metric for channel pruning The metrics for evaluating the importance of filters is critical. Existing CNN pruning methods design a variety of $\mathcal{L}(\cdot)$ on the filters. Consider the feature maps, contain rich and important information of both filter and input data, approaches using feature information have become popular and achieved state-of-the-art performance for channel pruning. The results of the feature maps may vary depending on the variability of the input data. Therefore, when the importance of one filter solely depends on the information represented by its own generated feature map, the measurement of the importance may be unstable and sensitive to the slight change of input data. So we have taken into account the characteristics of point clouds data and point-based networks architecture to improve the robustness of channel importance in point-based networks. On the one hand, we propose a coordinate-enhancement module by evaluating the correlation between the feature map and its corresponding points coordinates to guide point clouds network pruning, which will be described in Sec. 3.2. On the other hand, we design a knowledge recycling pruning schema, using discarded points to improve the channel importance evaluation criteria and increase the robustness of the pruning module, which will be described in detail in Sec. 3.3.

3.2. Coordinate-Enhanced Channel Importance

Dimensional information is critical in PNNs. The dimensional information (i.e., coordinates of the points) are usually adopted as input for feature extraction. Namely, the input and output of each SA block are concatenated with the coordinates of the points. Meanwhile, the intermediate feature maps reflect not only the information of the original input data but also the corresponding channel information. Therefore, the importance of the channel can be obtained from the feature maps, i.e., the importance of the corresponding channel. The dimensional information is crucial in point-based tasks and should be considered as part of importance metric. Thus the critical problem falls in designing a function that can well reflect the dimensional information richness of feature maps. The feature map, obtained by encoding points spatial x, y, and z coordinates, should be closely related to the original corresponding points coordinates. Therefore, we use the correlation between the current feature map and the corresponding input points coordinates to determine the importance of the filter. The designed Coordinate-Enhancement (CE) module based on Eq. (2):

$$\min_{\delta_{ij}} \sum_{i=1}^{K} \sum_{j=1}^{n_i} \delta_{ij} \mathcal{L}_c\left(\mathbf{F}_j^i\right), \text{ s.t. } \sum_{j=1}^{n_i} \delta_{ij} = k_l, \tag{3}$$

where δ_{ij} is an indicator which is 1 if $\mathbf{w}_j^i$ is to be pruned or 0 if $\mathbf{w}_j^i$ is to be kept and k_l is the kept channel number. $\mathcal{L}_c(\cdot)$ measures the importance of a channel from take account of the relationship between feature map and points coordinates which can be formulated as:

$$\mathcal{L}_c\left(F_j^i\right) = \mathcal{M}\{\text{CE}\left(F_j^i, x^i\right)\}, \tag{4}$$

where CE obtains the coordinate-enhanced score by calculating correlation of each channel in the feature map with

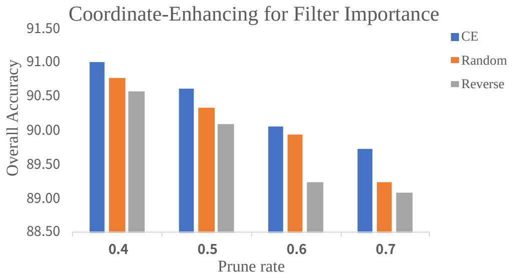

Figure 2. Comparisons on the pruned model accuracy with different pruning metrics with CE scores. Results are on ModelNet40 with PointNeXt-S (C=64).

the original coordinates, and $\mathcal{M}$ takes the maximum value. Hence, higher coordinate-enhanced score (i.e., $\mathcal{L}_c$) serve as a reliable measurement for information richness.

We evaluated the effectiveness of the CE module on ModelNet40 with PointNeXt-S (C=64). To demonstrate the experiment's validity, we compared the overall accuracy (OA) for pruning rates of 40%, 50%, 60%, and 70%, respectively. Three sets of experiments are carried out. First, the 'CE' group selected filters in order of value from the CE module. Secondly, the 'Random' group is used to randomly select filters for pruning, and finally, the 'Reverse' is used to select filters according to the coordinate-enhanced score from low to high. As shown in Fig. 2, the channel with a higher coordinate-enhanced score has higher accuracy, which means that the channel with a higher coordinate-enhanced score has higher importance and should be retained in the pruning process.

3.3. Knowledge Recycling

Sec. 3.2 shows that feature maps can reflect the importance of the corresponding channels. Another problem is that the importance determination may be unstable and sensitive to small changes in the input data as the feature maps are highly related to the samples of input data. Therefore, we aim to reduce the impact of such data variation. Through the analysis of the PNNs in Sec. 3.1, we found that some points are discarded to obtain hierarchical points set feature. These discarded points are informative as well since the sampling mechanism are highly random and can be leveraged to reduce the impact of data variation. The Knowledge Recycling (KR) module is proposed to reuse the discarded points to improve the robustness of channel pruning.

For those centroids that are computed in $(l-1)$-th layer but discarded in l-th layer due to sampling, which are equivalent to the sampled points. Therefore, the discarded centroids x_{dis} are feeded into l-th convolutional layer to generate the feature map $\mathbf{f_{dis}}^l$, and $\mathcal{L}_k\left(F^i_{dis_j}\right)$ is taken in use for the evaluation of channel importance.

We calculate the relevant feature maps from the network parameters trained from the sampled points and use them as part of the importance calculation for the current SA layer channels. Specifically, for each layer in the SA module, we obtain the features of the discard points by Eq. 1:

$$\mathcal{F}^l_{dis} = \left\{ \mathbf{f_{dis}}^l_1, \mathbf{f_{dis}}^l_2, \dots, \mathbf{f_{dis}}^l_{n_i} \right\} \in \mathbb{R}^{n_i\times(3+c_i)},$$

where n_i is sampled points number, c_i is the points feature dimension. So the supplement importance is:

$$\begin{aligned}\mathcal{L}_k\left(F^i_j\right) &= \mathcal{L}_k\left(F^i_{dis_j}\right)\\ &= \mathcal{M}\{\mathrm{CE}\left(F^i_j, x^i_{dis}\right)\},\end{aligned} \tag{5}$$

where x^i_{dis} are discard points for recycling.

It should be noted that the KR module only needs to calculate $\mathcal{F}^l_{dis}$ from the parameters trained by the sampled points and does not incur much additional overhead.

3.4. Using CP^3 in Pruning Methods

The overall CP^3 improve the existing CNNs pruning methods by considering the input data of PNNs and the PNN structure in Sec. 3.2 and Sec. 3.3, respectively. In fact, CP^3 can complement the existing pruning methods, i.e., as a plug-in to the existing pruning methods, to improve the pruning performance on PNNs. Specifically, combining Eq. (4) and Eq. (5), we obtain the final pruning formula according to Eq. (2):

$$\begin{aligned}&\min_{\delta_{ij}} \sum_{i=1}^{K}\sum_{j=1}^{n_i} \delta_{ij}(\mathcal{L}\left(\mathbf{w}^i_j\right) + \mathcal{L}_c\left(F^i_j\right) + \mathcal{L}_k\left(F^i_j\right)),\\ &\qquad\text{s.t.} \sum_{j=1}^{n_i} \delta_{ij} = k_l,\end{aligned} \tag{6}$$

where δ_{ij} is an indicator which is 1 if $\mathbf{w}^i_j$ is to be pruned or 0 if $\mathbf{w}^i_j$ is to be kept, $\mathcal{L}(\cdot)$ is original CNNs pruning method measure importance of a channel, $\mathcal{L}_c$ and $\mathcal{L}_k$ are coordinate-enhanced score and knowledge-recycling score, k_l is the kept channel number.

4. Experiments

4.1. Experimental Settings

Baseline models and datasets To demonstrate the effectiveness and generality of the proposed CP^3, we tested it on three different 3D tasks and five datasets with various PNNs and three recent advanced channel pruning methods. The evaluated pruning methods include HRank (*2020*) [24], ResRep (*2021*) [7], and CHIP (*2021*) [40]. For the *classification task*, we chose the classical PointNet++ [33] and PointNeXt [35] models as the original networks and conducted experiments on ModelNet40 [47] and ScanObjectNN [43]. Specifically, for PointNeXt-S we tested

Table 1. Comparisons of classification on the ModelNet40 [47] test set with PointNet++ [33], PointNeXt-S (C=32) [35], and PointNeXt-S (C=64). For PointNeXt-S (C=32), we report the baseline results from the original paper. For PointNet++ and PointNeXt-S (C=64), we report the baseline results obtained by OpenPoints [35] re-implementations trained with the improved strategies.

Method	**PointNet++**				**PointNeXt-S (C=32)**				**PointNeXt-S (C=64)**			
	OA	**mAcc**	**Params. (M)**	**GFLOPs (↓ %)**	**OA**	**mAcc**	**Params. (M)**	**GFLOPs (↓ %)**	**OA**	**mAcc**	**Params. (M)**	**GFLOPs (↓ %)**
Baseline	92.80	89.90	1.47	1.71 (–)	92.99	89.6	1.37	1.64 (–)	93.44	91.05	4.52	6.49 (–)
HRank	92.59	89.83	0.86	0.77 (55.0)	92.87	89.97	0.74	0.71 (56.7)	92.23	89.81	2.12	2.69 (56.7)
HRank +CP^3	92.95	89.91	0.84	0.75 (56.1)	93.23	90.56	0.71	0.67 (59.1)	93.52	90.33	2.01	2.58 (59.1)
HRank	91.79	88.82	0.59	0.42 (75.4)	92.63	89.12	0.50	0.39 (76.2)	92.71	90.45	1.33	1.56 (76.2)
HRank +CP^3	92.54	88.52	0.57	0.39 (77.2)	93.03	90.92	0.49	0.38 (76.8)	93.07	90.55	1.28	1.50 (76.8)
HRank	91.34	88.18	0.36	0.15 (91.2)	92.73	89.98	0.43	0.29 (82.3)	92.83	89.99	0.74	0.71 (82.3)
HRank +CP^3	91.71	88.68	0.34	0.13 (92.4)	92.99	90.11	0.40	0.27 (83.5)	93.11	90.52	0.71	0.67 (83.5)
ResRep	92.71	90.39	0.85	0.73 (57.3)	92.83	90.15	0.81	0.69 (57.9)	91.61	88.42	2.08	2.34 (57.9)
ResRep+CP^3	93.27	90.48	0.82	0.70 (59.1)	93.35	90.93	0.79	0.67 (59.1)	92.93	90.66	1.89	2.02 (59.1)
ResRep	92.50	89.25	0.57	0.41 (76.0)	92.64	90.01	0.51	0.40 (75.6)	91.67	89.27	1.13	1.92 (75.6)
ResRep+CP^3	92.46	89.43	0.54	0.40 (76.6)	93.41	90.87	0.49	0.38 (76.8)	93.11	90.82	1.02	1.89 (76.8)
ResRep	92.11	89.00	0.55	0.24 (86.0))	92.30	89.31	0.34	0.21 (87.2)	89.54	88.54	0.72	0.69 (87.2)
ResRep+CP^3	92.48	89.21	0.58	0.21 (87.7)	92.95	90.70	0.33	0.18 (89.0)	91.02	89.82	0.69	0.65 (89.0)
CHIP	92.79	89.23	0.82	0.73 (57.3)	93.11	90.27	0.71	0.67 (59.1)	93.03	90.60	1.48	1.44 (59.1)
CHIP+CP^3	92.99	90.66	0.81	0.70 (59.1)	93.35	90.80	0.69	0.65 (60.4)	93.35	91.11	1.45	1.40 (60.4)
CHIP	92.45	89.19	0.57	0.39 (77.2)	92.71	89.90	0.49	0.38 (76.8)	92.79	90.39	0.92	0.76 (76.8)
CHIP+CP^3	92.91	89.65	0.54	0.35 (79.5)	93.03	90.51	0.48	0.37 (77.4)	93.23	90.30	0.89	0.74 (77.4)
CHIP	92.26	89.56	0.36	0.15 (91.2)	92.42	89.13	0.32	0.16 (90.2)	90.83	88.70	0.65	0.46 (90.2)
CHIP+CP^3	92.71	90.41	0.34	0.13 (92.4)	92.50	90.35	0.30	0.14 (91.5)	92.87	90.25	0.63	0.44 (91.5)

two settings with widths of 32 and 64. For the *segmentation* task, we conducted experiments on S3DIS [1] with PointNeXt-B and PointNeXt-L [35] as the original PNNs. For the *object detection* task, we pruned two point-based detectors (VoteNet [31] and GroupFree3D [25]) on SUN RGB-D [38] and ScanNetV2 [5].

Implementation details We conducted the classification and segmentation experiments with OpenPoints [35] and the object detection experiments with MMdetection3D [3], all on NVIDIA P100 GPUs. For a fair comparison, we used the same hyperparameter settings for each group of experiments. We either 1) measured the parameter/FLOP reductions of the pruned networks with similar performance or 2) measured the performance of the pruned networks with a similar amount of parameter/FLOP reductions. For all experiments, we reported the number of FLOPs ('GFLOPs') and parameters ('Params.'), as well as task-specific metrics to be described in each experiment. More experimental results are available in the supplementary.

4.2. Results on Classification

ModelNet40 ModelNet40 [47] contains 9843 training and 2468 testing meshed CAD models belonging to 40 categories. Following the standard practice [33], we report the class-average accuracy (mAcc) and the overall accuracy (OA) on the testing set. We compared the pruned networks directly by HRank, ResRep, and CHIP and those pruned by the three pruning methods with CP^3. As shown in Tab. 1, our CP^3 improved the performance of existing CNNs pruning methods for different PNNs with various pruning rates. CP^3 had higher accuracy scores with a similar (and mostly higher) pruning rates. Notably, with the pruning rate of 58%, CP^3 usually produced compact PNNs with even better accuracy scores than the original PNNs, which is difficult for pruning methods without CP^3.

ScanObjectNN We also conducted experiments on the ScanObjectNN benchmark [43]. ScanObjectNN contains 15000 objects categorized into 15 classes with 2902 unique object instances in the real world. As reported in Tab. 2, CP^3 surpassed existing CNN pruning methods directly appled to PNNs. For example, comparing with the baseline pruning method HRank, CP^3 boosts the OA score of PointNet++ by 1.14% (85.01→86.15), 1.14% (84.94→86.08), and 0.59% (84.03→84.62) for the three different pruning rates. Similarly, CP^3 obtains much higher OA and mAcc scores than the three baselines with different pruning rates on PointNet++ and PointNeXt-S. With the extensive experimental results on classification tasks, we show that CP^3 surely improved the pruned network's performance to a distinct extent compared to direct applications of 2D CNN pruning methods.

4.3. Results on Semantic Segmentation

S3DIS [1] is a challenging benchmark composed of 6 large-scale indoor areas, 271 rooms, and 13 semantic categories in total. Following a common protocol [41], we evaluated the presented approach in Area-5, which means to test on Area-5 and to train on the rest. For evaluation metrics, we used the mean classwise intersection over union (mIoU), the mean classwise accuracy (mAcc), and the overall point-

Table 2. Comparisons of classification on the ScanObjectNN [43] test set with PointNet++ [33], PointNeXt-S (C=32) [35], and PointNeXt-S (C=64). For PointNeXt-S (C=32), we report the baseline results from the original paper. For PointNet++ and PointNeXt-S (C=64), we report the baseline results obtained by OpenPoints [35] re-implementations trained with the improved strategies.

Method	**PointNet++**				**PointNeXt-S (C=32)**				**PointNeXt-S (C=64)**			
	OA	**mAcc**	**Params. (M)**	**GFLOPs (↓ %)**	**OA**	**mAcc**	**Params. (M)**	**GFLOPs (↓ %)**	**OA**	**mAcc**	**Params. (M)**	**GFLOPs (↓ %)**
Baseline	86.20	84.40	1.47	1.71 (–)	87.40	85.39	1.37	1.64 (–)	88.20	86.84	4.52	6.49 (–)
HRank	85.01	83.33	0.82	0.73 (57.3)	87.02	85.85	0.72	0.69 (57.9)	87.51	85.48	2.22	2.91 (55.2)
HRank+CP^3	86.15	84.40	0.80	0.70 (59.1)	87.47	86.21	0.70	0.67 (59.1)	87.95	86.02	2.16	2.82 (56.5)
HRank	84.94	83.43	0.59	0.43 (74.9)	84.79	81.93	0.50	0.39 (76.2)	84.66	81.00	1.33	1.56 (76.0)
HRank+CP^3	86.08	84.51	0.59	0.41 (76.0)	86.40	83.94	0.48	0.37 (77.4)	86.43	84.94	1.28	1.50 (76.9)
HRank	84.03	82.16	0.37	0.16 (90.6)	81.33	78.32	0.32	0.17 (89.6)	85.39	83.83	0.73	0.71 (89.1)
HRank+CP^3	84.62	82.29	0.36	0.15 (91.2)	84.21	82.13	0.31	0.16 (90.2)	86.26	84.36	0.70	0.67 (89.7)
ResRep	86.77	84.78	0.82	0.76 (55.6)	86.32	84.45	0.69	0.65 (60.4)	86.58	84.02	2.12	2.82 (56.5)
ResRep+CP^3	87.11	85.50	0.82	0.71 (58.5)	86.66	84.75	0.67	0.64 (61.0)	88.52	86.00	2.03	2.74 (57.8)
ResRep	83.99	82.34	0.60	0.44 (74.3)	85.28	83.27	0.48	0.38 (76.8)	85.22	82.03	1.39	1.65 (74.6)
ResRep+CP^3	84.32	83.68	0.59	0.43 (74.9)	86.22	84.32	0.47	0.37 (77.4)	86.73	84.23	1.32	1.54 (76.3)
ResRep	83.79	81.91	0.40	0.29 (83.0)	83.66	81.66	0.41	0.25 (84.8)	85.02	81.27	0.60	0.68 (89.5)
ResRep+CP^3	84.80	82.83	0.40	0.26 (84.8)	85.01	83.03	0.40	0.23 (86.0)	86.13	83.40	0.58	0.52 (92.0)
CHIP	86.14	84.98	0.82	0.73 (57.3)	87.13	85.09	0.70	0.67 (59.1)	88.45	87.40	2.11	2.74 (57.8)
CHIP+CP^3	86.25	84.68	0.80	0.70 (59.1)	87.54	85.67	0.69	0.65 (60.4)	88.58	86.45	2.05	2.65 (59.2)
CHIP	84.45	82.25	0.59	0.42 (75.4)	85.28	83.28	0.50	0.4 (75.6)	86.68	85.12	1.37	1.64 (74.7)
CHIP+CP^3	85.59	84.42	0.57	0.40 (76.6)	86.29	83.99	0.49	0.38 (76.8)	87.79	86.81	1.32	1.56 (76.0)
CHIP	83.27	81.15	0.36	0.15 (91.2)	81.37	78.99	0.34	0.19 (88.4)	83.90	81.83	0.44	0.32 (95.1)
CHIP+CP^3	84.03	82.22	0.35	0.14 (91.8)	82.12	79.41	0.33	0.18 (89.0)	84.25	82.17	0.42	0.29 (95.5)

Table 3. Comparisons of semantic segmentation on the S3DIS dataset (evaluated in Area-5) with PointNeXt-B and PointNeXt-L [35].

Method	**PointNeXt-B**					**PointNeXt-L**				
	OA	**mAcc**	**mIoU**	**Params. (M)**	**GFLOPs (↓ %)**	**OA**	**mAcc**	**mIoU**	**Params. (M)**	**GFLOPs (↓ %)**
Baseline	89.40	73.90	67.50	3.83	8.80 (–)	90.10	75.70	69.30	7.13	15.24 (–)
HRank	89.04	72.14	65.66	1.72	4.01 (54.4)	88.88	73.61	66.80	3.20	6.83 (55.2)
HRank+CP^3	89.24	73.76	66.95	1.61	3.80 (56.8)	89.44	74.27	67.53	3.00	6.48 (57.5)
HRank	88.81	72.16	65.58	0.85	2.04 (76.8)	88.21	70.93	64.30	1.58	3.47 (77.2)
HRank+CP^3	89.02	72.82	66.41	0.78	1.85 (78.9)	88.35	71.26	64.71	1.44	3.14 (79.4)
CHIP	88.89	73.26	66.57	1.66	3.93 (55.3)	89.16	73.72	67.09	3.09	6.68 (56.2)
CHIP+CP^3	89.68	73.14	66.80	1.56	3.65 (58.5)	89.47	74.20	67.28	2.91	6.22 (59.2)
CHIP	88.67	73.24	66.68	0.81	1.92 (78.2)	88.58	71.58	65.18	1.50	3.27 (78.5)
CHIP+CP^3	89.81	73.36	66.95	0.74	1.79 (79.7)	89.20	71.66	65.24	1.38	3.04 (80.1)

Table 4. Comparisons of object detection on the ScanNet dataset.

Method	**mAP@0.25**	**mAP@0.50**	**Params. (K)**	**GFLOPs (↓%)**
Baseline (VoteNet)	62.34	40.82	641.92	5.78 (–)
ResRep	62.45	40.95	251.23	2.45 (57.6)
ResRep+CP^3	**63.92**	**41.47**	**242.26**	**2.41 (58.1)**
ResRep	61.78	40.54	180.49	1.83 (68.3)
ResRep+CP^3	**62.98**	**40.94**	**160.48**	**1.78 (69.2)**

Table 5. Comparisons of object detection on SUN RGB-D.

Method	**mAP@0.25**	**mAP@0.50**	**Params. (K)**	**GFLOPs (↓%)**
Baseline (VoteNet)	59.78	35.77	641.92	5.78 (–)
ResRep	59.37	36.80	179.91	2.42 (58.13)
ResRep+CP^3	**60.10**	**37.37**	**172.35**	**2.20 (61.93)**
ResRep	59.01	35.91	135.13	1.84 (68.17)
ResRep+CP^3	**59.18**	**36.24**	**129.64**	**1.83 (68.34)**

wise accuracy (OA). As the segmentation task is relatively difficult and the segmentation network structures are relatively complex, we pruned only the encoder part of the network and kept the original decoder part. The results are presented in Tab. 3. As expected, the performance of the pruned networks degraded more from the original networks than those in the classification experiments, but the performance was acceptable. Meanwhile, in all cases, with our CP^3, PNNs have a higher accuracy at a higher pruning rate than without CP^3. For example, for PointNeXt-B, comparing with directly applying CHIP, incorporating CP^3 obtained much higher OA, mAcc and mIoU scores (1.2% on OA and 0.3% on mIoU). The results about segmentation have well validated the generalization of CP^3 to new and difficult tasks.

4.4. Results on 3D Object Detection

4.4.1 Evaluation and Comparison of VoteNet

Tabs. 4 and 5 show the results of the pruned VoteNet models on the ScanNetV2 and SUN RGB-D datasets, respectively. We evaluated the performance of our proposed method in

Table 6. Comparisons of object detection on the ScanNet dataset. The baseline PNN model is GroupFree3D.

Method	**mAP@0.25**	**mAP@0.50**	**Params. (K)**	**GFLOPs (↓%)**
Baseline	68.22	52.61	2438.34	21.78 (–)
ResRep	68.24	51.48	1910.73	10.90 (49.95)
ResRep+CP^3	**68.86**	**52.08**	**1654.35**	**10.88 (50.05)**
ResRep	67.21	51.28	1703.34	8.71 (60.01)
ResRep+CP^3	**68.57**	**51.85**	**1501.46**	**8.68 (60.14)**

terms of the mean average precision at IOU threshods of 0.25 and 0.50 (mAP@25 and mAP@50).

ScanNetV2 ScanNetV2 [5] is a richly annotated dataset of 3D reconstructed meshes of indoor scenes. It contains about 1200 training examples collected from hundreds of different rooms and is annotated with semantic and instance segmentation for 18 object categories. Tab. 4 shows the results of directly applying ResRep and with CP^3. As can be seen from the table, the accuracy of the 2D method directly applied to the 3D network decreased by a flops drop of about 60%, while our method achieves 1.58% and 0.65% improvements at mAP@0.25 and mAP@0.5 with a drop rate of 58.13% FLOPS. When FLOPs drop to about 70%, the accuracy of the direct porting CNNs pruning method works poorly, while the improvement of mAP@0.25 and mAP@0.5 of our method is 0.64% and 0.12%, respectively.

SUN RGB-D The SUN RGB-D dataset [38] consists of 10355 single-view indoor RGB-D images annotated with over 64000 3D bounding boxes and semantic labels for 37 categories. We conducted experiments on SUN RGB-D with the same setup as those on ScanNetV2. The findings are also similar to those in ScanNetV2. It can be observed from Tab. 5 that the accuracy of directly transplanted CNNs pruning method is reduced to some extent (mAP@0.25 reduced by 0.41) when FLOPs drop by 58.13%, while CP^3 improved the detection model by 0.32% mAP@0.25 and 1.60% mAP@0.5 with a FLOPs drop of 61.93%. Even when FLOPs drop to 70%, our method's mAP@0.25 drop only 0.6%, which is obviously better.

4.4.2 Evaluation and Comparison on GroupFree3D

We also conducted experiments on another point-based 3D detection model, GroupFree3D, on ScanNetV2. Tab. 6 summarizes the pruning performance of our approach for GroupFree3D on the ScanNetV2 dataset. When targeting a moderate compression ratio, our approach can achieve 32.15% and 50.05% storage and computation reductions, respectively, with a 0.64% accuracy increase for mAP@0.25 over the baseline model. In the case of higher compression ratio, CP^3 still achieves superior performance to other methods. Specifically, the ResRep loses accuracy by 1.01% mAP@0.25 when the parameters and flops drop by 30.14% and 60.01%, while in our method, the accuracy increases 0.35% for mAP@0.25.

Table 7. Ablation study of different components in CP^3. Results are of classification on the ScanObjectNN dataset with PointNeXt-S (C=32) as the baseline. 'CE' represents the coordinate-enhanced module, and 'KR' represents the knowledge recycling module.

Setting	**CE**	**KR**	**Pruning Rate**	**OA**	**mAcc**
Baseline			–	88.20	86.40
HRank			0.75	84.79	81.93
		✓	0.75	85.63	82.97
	✓		0.75	85.11	82.13
HRank+CP^3	✓	✓	0.75	**86.63**	**83.63**
HRank			0.90	81.33	78.32
		✓	0.90	83.66	81.32
	✓		0.90	83.10	80.47
HRank+CP^3	✓	✓	0.90	**84.83**	**82.74**

4.5. Ablation Studies

We conducted ablation studies to validate the Coordinate-Enhancement (CE) module and the Knowledge-Recycling (KR) module in CP^3. All results provided in the section are tested on ScanObjectNN with PointNeXt-S (C=32) as the baseline and HRank as the pruning method. We evaluated the pruned networks' performance with the FLOPs drop of 75% and 90%, and with/without the CE and KR modules to the pruning method. From Tab. 7, we find that KR improves OA of 0.84% and 2.33% when pruning rate is 75% and 90%, respectively, while CE improves OA of 0.32% and 1.77% at 75% and 90% pruning rates, respectively. Bringing the two modules together, the OA improvement is 1.84% and 3.50% at 75% and 90% pruning rates, respectively. The ablation study results have validated the effectiveness of all designs in CP^3.

5. Conclusion

In this paper, we focus on 3D point-based network pruning and design a 3D channel pruning plug-in (CP^3) that can be used with existing 2D CNN pruning methods. To the best of our knowledge, this is the first pruning work explicitly considering the characteristics of point cloud data and point-based networks. Empirically, we show that the proposed CP^3 is universally effective for a wide range of point-based networks and 3D tasks.

Acknowledgement

This work was partially supported by National Natural Science Foundation of China (NSFC) under Grant 62271203, the Science and Technology Innovation Action Plan of Shanghai under Grant 2251110540, and Shanghai Pujiang Program under Grant 21PJ1420300.

References

[1] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis K. Brilakis, Martin Fischer, and Silvio Savarese. 3D semantic parsing of large-scale indoor spaces. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1534–1543, 2016. 6, 12

[2] Chen Chen, Zhe Chen, Jing Zhang, and Dacheng Tao. SASA: semantics-augmented set abstraction for point-based 3D object detection. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI*, pages 221–229, 2022. 1

[3] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020. 6

[4] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 12

[5] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2432–2443, 2017. 6, 8

[6] Xiaohan Ding, Guiguang Ding, Yuchen Guo, and Jungong Han. Centripetal sgd for pruning very deep convolutional networks with complicated structure. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4943–4953, 2019. 1, 3

[7] Xiaohan Ding, Tianxiang Hao, Jianchao Tan, Ji Liu, Jungong Han, Yuchen Guo, and Guiguang Ding. ResRep: Lossless CNN pruning via decoupling remembering and forgetting. In *IEEE/CVF International Conference on Computer Vision, ICCV*, pages 4510–4520, 2021. 1, 3, 5

[8] Anand Dubey, Avik Santra, Jonas Fuchs, Maximilian Lübke, Robert Weigel, and Fabian Lurz. Haradnet: Anchor-free target detection for radar point clouds using hierarchical attention and multi-task learning. *Machine Learning with Applications*, 8:100275, 2022. 1

[9] Hamidreza Fazlali, Yixuan Xu, Yuan Ren, and Bingbing Liu. A versatile multi-view framework for LiDAR-based 3D object detection with guidance from panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 17171–17180, 2022. 1

[10] Kehong Gong, Bingbing Li, Jianfeng Zhang, Tao Wang, Jing Huang, Michael Bi Mi, Jiashi Feng, and Xinchao Wang. PoseTriplet: Co-evolving 3d human pose estimation, imitation, and hallucination under self-supervision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 11017–11027, 2022. 1

[11] Google. Tensorflow lite. https://www.tensorflow.org/lite, 2017. 2

[12] Yushuo Guan, Ning Liu, Pengyu Zhao, Zhengping Che, Kaigui Bian, Yanzhi Wang, and Jian Tang. DAIS: Automatic channel pruning via differentiable annealing indicator search. *IEEE Transactions on Neural Networks and Learning Systems, TNNLS*, pages 1–12, 2022. 1, 3

[13] Kai Han, Yunhe Wang, Qiulin Zhang, Wei Zhang, Chunjing Xu, and Tong Zhang. Model rubik's cube: Twisting resolution, depth and width for tinynets. *Advances in Neural Information Processing Systems*, 33:19353–19364, 2020. 3

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF conference on Computer Vision and Pattern Recognition, CVPR*, pages 770–778, 2016. 1

[15] Zejiang Hou, Minghai Qin, Fei Sun, Xiaolong Ma, Kun Yuan, Yi Xu, Yen-Kuang Chen, Rong Jin, Yuan Xie, and Sun-Yuan Kung. CHEX: channel exploration for CNN model compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 12277–12288, 2022. 3

[16] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. 1

[17] Fengwei Jia, Xuan Wang, Jian Guan, Huale Li, Chen Qiu, and Shuhan Qi. Arank: Toward specific model pruning via advantage rank for multiple salient objects detection. *Image and Vision Computing*, 111:104192, 2021. 12

[18] Xiaotang Jiang, Huan Wang, Yiliu Chen, Ziqi Wu, Lichuan Wang, Bin Zou, Yafeng Yang, Zongyang Cui, Yu Cai, Tianhang Yu, et al. MNN: A universal and efficient inference engine. *The Conference on Machine Learning and Systems, MLSys*, pages 1–13, 2020. 3

[19] Hema Koppula, Abhishek Anand, Thorsten Joachims, and Ashutosh Saxena. Semantic labeling of 3d point clouds for indoor scenes. *Advances in Neural Information Processing Systems, NeurIPS*, pages 244–252, 2011. 2

[20] Bastian Leibe, Aleš Leonardis, and Bernt Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1):259–289, 2008. 3

[21] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *International Conference on Learning Representations, ICLR*, 2017. 3

[22] Kelin Li, Nicholas Baron, Xian Zhang, and Nicolas Rojas. Efficientgrasp: A unified data-efficient learning to grasp method for multi-fingered robot hands. *IEEE Robotics and Automation Letters*, 7(4):8619–8626, 2022. 1

[23] Lucas Liebenwein, Cenk Baykal, Harry Lang, Dan Feldman, and Daniela Rus. Provable filter pruning for efficient neural networks. In *8th International Conference on Learning Representations, ICLR*, 2020. 3

[24] Mingbao Lin, Rongrong Ji, Yan Wang, Yichen Zhang, Baochang Zhang, Yonghong Tian, and Ling Shao. HRank: Filter pruning using high-rank feature map. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1529–1538, 2020. 1, 3, 4, 5

[25] Liyang Liu, Shilong Zhang, Zhanghui Kuang, Aojun Zhou, Jing-Hao Xue, Xinjiang Wang, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Group Fisher pruning for practical network compression. In *International Conference on Machine Learning, ICML*, pages 7021–7032, 2021. 6

[26] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *IEEE/CVF International Conference on Computer Vision, ICCV*, pages 2736–2744, 2017. 1, 3

[27] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3D object detection via transformers. In *IEEE/CVF International Conference on Computer Vision, CVPR*, pages 2949–2958, 2021. 3

[28] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. In *International Conference on Learning Representations, ICLR*, 2022. 3

[29] Fanxu Meng, Hao Cheng, Ke Li, Huixiang Luo, Xiaowei Guo, Guangming Lu, and Xing Sun. Pruning filter in filter. In *Advances in Neural Information Processing Systems, NeurIPS*, volume 33, pages 17629–17640, 2020. 1, 3

[30] Quang-Hieu Pham, Thanh Nguyen, Binh-Son Hua, Gemma Roig, and Sai-Kit Yeung. JSIS3D: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2019. 1

[31] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3D object detection in point clouds. In *IEEE/CVF International Conference on Computer Vision, ICCV*, 2019. 3, 6, 12

[32] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 652–660, 2017. 3

[33] 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, NeurIPS*, pages 5100–5109, 2017. 3, 5, 6, 7

[34] Guocheng Qian, Hasan Hammoud, Guohao Li, Ali Thabet, and Bernard Ghanem. ASSANet: An anisotropic separable set abstraction for efficient point cloud representation learning. In *Advances in Neural Information Processing Systems, NeurIPS*, pages 28119–28130, 2021. 3

[35] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. PointNeXt: Revisiting pointnet++ with improved training and scaling strategies. In *Advances in Neural Information Processing Systems, NeurIPS*, 2022. 3, 5, 6, 7, 12, 13

[36] Ruizhi Shao, Zerong Zheng, Hongwen Zhang, Jingxiang Sun, and Yebin Liu. DiffuStereo: High quality human reconstruction via diffusion-based stereo using sparse cameras. In *European Conference on Computer Vision, ECCV*, pages 702–720, 2022. 1

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations, ICLR*, 2015. 1, 2

[38] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun RGB-D: A RGB-D scene understanding benchmark suite. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 567–576, 2015. 6, 8

[39] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun RGB-D: A RGB-D scene understanding benchmark suite. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 567–576, 2015. 12

[40] Yang Sui, Miao Yin, Yi Xie, Huy Phan, Saman Aliari Zonouz, and Bo Yuan. CHIP: Channel independence-based pruning for compact neural networks. In *Advances in Neural Information Processing Systems, NeurIPS*, pages 24604–24616, 2021. 1, 3, 4, 5, 12

[41] Lyne P. Tchapmi, Christopher B. Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. SEGCloud: Semantic segmentation of 3d point clouds. In *International Conference on 3D Vision, 3DV*, pages 537–547, 2017. 6

[42] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. KpConv: Flexible and deformable convolution for point clouds. In *IEEE/CVF international conference on computer vision, CVPR*, pages 6411–6420, 2019. 3

[43] 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. In *IEEE/CVF International Conference on Computer Vision, ICCV*, pages 1588–1597, 2019. 5, 6, 7

[44] Haowen Wang, Mingyuan Wang, Zhengping Che, Zhiyuan Xu, Xiuquan Qiao, Mengshi Qi, Feifei Feng, and Jian Tang. RGB-Depth fusion GAN for indoor depth completion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 6199–6208, 2022. 1

[45] Wenxiao Wang, Minghao Chen, Shuai Zhao, Long Chen, Jinming Hu, Haifeng Liu, Deng Cai, Xiaofei He, and Wei Liu. Accelerate cnns from three dimensions: A comprehensive pruning framework. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 10717–10726. PMLR, 2021. 3

[46] Wenxiao Wang, Cong Fu, Jishun Guo, Deng Cai, and Xiaofei He. COP: customized deep model compression via regularized correlation-based filter-level pruning. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 3785–3791. ijcai.org, 2019. 3

[47] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1912–1920, 2015. 5, 6

[48] Chenfeng Xu, Shijia Yang, Tomer Galanti, Bichen Wu, Xiangyu Yue, Bohan Zhai, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Image2point: 3d point-cloud understanding with 2d image pretrained models. In

European Conference on Computer Vision, pages 638–656. Springer, 2022. 2

[49] Wei Yang, Chris Paxton, Maya Cakmak, and Dieter Fox. Human grasp classification for reactive human-to-robot handovers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 11123–11130, 2020. 1

[50] Xucheng Ye, Pengcheng Dai, Junyu Luo, Xin Guo, Yingjie Qi, Jianlei Yang, and Yiran Chen. Accelerating CNN training by pruning activation gradients. In *European Conference on Computer Vision, ECCV*, pages 322–338, 2020. 3

[51] Ruichi Yu, Ang Li, Chun-Fu Chen, Jui-Hsin Lai, Vlad I Morariu, Xintong Han, Mingfei Gao, Ching-Yung Lin, and Larry S Davis. Nisp: Pruning networks using neuron importance score propagation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 9194–9203, 2018. 3

[52] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point transformer. In *IEEE/CVF International Conference on Computer Vision, ICCV*, pages 16239–16248, 2021. 3

[53] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. SE-SSD: self-ensembling single-stage object detector from point cloud. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 14494–14503, 2021. 1