Let 2D Images Give You More: Point Cloud Cross-Modal Training for Shape Classification

Anonymous Author(s)
Affiliation
Address
email

Abstract

Although recent point cloud analysis achieves impressive progress, the paradigm of representation learning from a single modality gradually meets its bottleneck. In this work, we take a step towards more discriminative 3D point cloud representation by fully taking advantages of 2D images which inherently contain richer appearance information, *e.g.,* texture, color, and shade. Specifically, this paper introduces a simple but effective point cloud cross-modality training (**PointCMT**) strategy, which utilizes view-images, *i.e.*, rendered or projected 2D images of the 3D object, to boost point cloud classification. In practice, to effectively acquire auxiliary knowledge from view images, we develop a teacher-student framework and formulate the cross-modal learning as a knowledge distillation problem. PointCMT eliminates the distribution discrepancy between different modalities through novel feature and classifier enhancement criteria and avoids potential negative transfer effectively. Note that PointCMT effectively improves the point-only representation without any architecture modification. Sufficient experiments verify significant gains on various datasets using appealing backbones, *i.e.*, equipped with PointCMT, PointNet++ and PointMLP achieve state-of-the-art performance on two benchmarks, *i.e.,* **94.4%** and **86.6%** accuracy on ModelNet40 and ScanObjectNN, respectively.

1 Introduction

As the fundamental 3D representation, point clouds have attracted increasing attention for various applications, *e.g.*, self-driving [3, 35, 36], robotics perception [9, 15, 8], *etc*. Generally, a point cloud consists of sparse and unordered points in the 3D space, which is significantly different from a 2D image with a dense and regular pixel array. Prior studies treat the understanding of 2D images and 3D point clouds as two separate problems, and both have their own merits and drawbacks. Concretely, rich color and fine-grained texture are easily obtained in 2D images, but they are ambiguous in depth and shape sensing. Previous works extract features on images through convolution neural networks (CNN). In contrast, point clouds are superior in providing spatial and geometric information but only preserve sparse and textureless features. Several prior studies process features on unstructured point clouds through local aggregation operators [36, 46]. It is natural to raise a question: *could we use the rich information hidden in 2D images to boost 3D point cloud shape analysis*?

To address the above issue, one straightforward way is to leverage the benefits of both images and point clouds, *i.e.*, fusing information from two complementary representations with task-specific design [60, 26, 12, 34, 45]. However, utilizing additional image representation requires designing a multi-modal network, which takes the extra image inputs in both training and inference phases. Moreover, the exploiting extra-images is usually computation-intensive and paired-images are usually unavailable during inference. Thus, multi-modal learning meets its bottleneck in many aspects.

This paper tries to ease the barrier of cross-modal learning between images and point clouds. Inspired by knowledge distillation (KD) that achieves knowledge transfer from a teacher model to a student

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

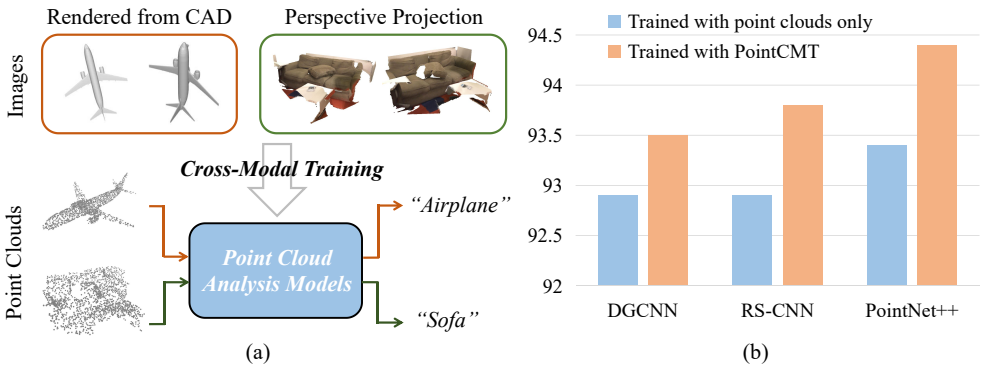

Figure 1: (a) Our proposed general Cross-Modal Training (PointCMT) strategy. It introduces priors from images into point cloud shape analysis models only in the training stage without any baseline model modification. (b) Classification accuracy (%) on ModelNet40 with or without training with our proposed PointCMT strategy. Noticeable improvements can be observed.

one, we formulate the cross-modal learning as a KD problem, conducting alignment between sample representations learned by images and point clouds. However, previous KD approaches usually assume that the training data used by the teacher and student are from the same distribution [19]. Thus, since sparse and disordered point clouds represent visual information different from images, naive feature alignment between two representations appeals to cause limited gains or negative transfer for the cross-modal scenario. To this end, we design a novel framework for cross-modal KD and propose the point cloud cross-modal training strategy, *i.e.*, **PointCMT** in Figure 1 (a), which distills features derived from images into the point cloud representation. Specifically, multiple view images for each 3D object can be generated through either rendering the CAD model or conducting perspective projection on the point cloud from different viewpoints. These free auxiliary images are fed into the image network to obtain the global representations for the object. Besides, feature and classifier enhancements are conducted between the point cloud and image features, in which the newly proposed criteria effectively avoid negative transfer between different modalities, *i.e.*, directly applying [19] hampers the performance on ModelNet40. After training, the model gains higher performance, only taking point clouds as input without any architecture modification.

Compared with multi-modal approaches, our solution has the following preferable properties: **1) Generality**: It can be easily integrated with arbitrary point cloud analysis models without structural modification. **2) Effectively**: It significantly boosts the performance upon several baseline approaches, *e.g.*, PointNet++ [36] achieves state-of-the-art 94.4% from 93.4% overall accuracy on ModelNet40, as shown in Figure 1 (b). **3) Efficiency**: Our PointCMT only utilizes auxiliary image data in the training stage. After training, the enhanced 3D model can be deployed without image inputs. **4) Flexibility**: The extensive experiments illustrate that PointCMT performs superior even without colorized and dedicated rendered images, *i.e.*, it can also greatly improve the performance when using images directly projected by sparse and textureless point clouds. Thus, it provides an alternative solution to enhance the point cloud shape analysis when the additional rendered images are not accessible.

In summary, our contributions are: **1)** This paper formulates cross-modal learning on point cloud analysis as a knowledge distillation problem, where we utilize the merits of texture and color-aware 2D images to acquire more discriminative point cloud representation. **2)** We propose point cloud cross-modal training, *i.e.,* PointCMT, strategy with corresponding criteria to boost point cloud models during the training stage. **3)** Extensive experiments on several datasets verify the effectiveness of our approach, where PointCMT greatly boosts several baseline models even on the state-of-the-art, *e.g.*, PointNet++ [36] trained with PointCMT gains **1.0%** and **4.4%** accuracy improvements on ModelNet40 [51] and ScanObjectNN, respectively. Even based upon PointMLP [33], it increases its accuracy by 1% to **86.7%** on ScanObjectNN dataset.

2 Related Works

3D shape recognition based on point clouds. These stream methods directly process raw point clouds as input (also called *point-based methods*). They are pioneered by PointNet [35], which

approximates a permutation-invariant set function using a per-point Multi-Layer Perceptron (MLP) followed by a max-pooling layer. Later on, point-based methods aim at designing local aggregation operators for local feature extraction. Specifically, they generally sample multiple sub-points from the original point cloud, and then aggregate neighboring features of each sub-point through local aggregation operators, in which point-wise MLPs [36, 33], adaptive weight [47, 49, 29] and pseudo grid based methods [42, 22] are proposed. More recently, there are some attempts to utilize non-local operator [56], or Transformer [64, 16] to mine the long distance dependency. This paper also follows the paradigm of point-based methods to conduct point cloud shape analysis.

3D shape recognition based on images. Since point clouds are irregular and unordered, some works consider projecting the 3D shapes into multiple images from different viewpoints (also called *view-based methods*) and then leverage the well-developed 2D CNNs to process 3D data. One seminal work of multi-view learning is MVCNN [41]. It extracts per-view features with a shared CNN in parallel, then aggregates via a view-level max-pooling layer. Most follow-up works propose more effective modules to aggregate the view-level features. For instance, some of them enhance the aggregated feature by considering similarity among views [13, 58] while others focus on the viewpoint relation [48, 24]. The above methods usually utilize ad-hoc rendered images for each 3D shape, including shade and texture for the surface mesh. Therefore, they generally achieve higher performance than point-based methods using sparse point clouds as input. Recently, [14] proposes a simple but effective method (*SimpleView*) through directly projecting sparse point clouds onto image planes, achieving comparable performance with point-based methods. Inspired by view-based methods, this paper takes advantage of extracted image features from the view-based method, which are utilized as prior knowledge to boost point cloud shape analysis.

Knowledge distillation. Knowledge distillation (KD) aims at compressing a large network (teacher) to a compact and tiny network (student) through knowledge transfer, and the student obtains a performance boost at the same time. The concept was first shown by Hinton *et al.* [19], which trains a student by using the softened logits of a teacher as targets. Subsequent research [2, 5] enhanced distillation by matching intermediate representations in the networks along with outputs using different approaches. Zagoruyko *et al.* [62] proposed to align attentional activation maps between networks. Srinivas and Fleuret [40] improved it by applying Jacobian matching to networks. Similarly, other recent approaches [25, 1, 7, 43, 11, 65] use different criteria to align the sample representations between the teacher and student. However, almost all the existing works assume that the training data used by the teacher and student networks are from the same distribution. Our experiment illustrates that new biases and negative transfer will be introduced in the distillation process if cross-modal data from different distribution (*e.g.*, features extracted from unordered point cloud and regular grid image) is utilized directly on traditional KD approaches.

Cross-modal knowledge transfer. Cross-modal knowledge transfer in computer vision is a relatively emerging field that aims to only utilize additional modalities at the training stage and enhance the model's performance on the target modality at the inference stage. Recently, there are 3D-to-2D knowledge transfer approaches, which adopt geometric aware 3D features from point clouds to enhance the performance of 2D tasks through a contrastive manner [20] or feature alignment [32]. Later on, works attempt to transfer priors in images to enhance 3D point cloud-related tasks, and some are designed for specific tasks. Among them, the images-to-point contrastive pre-training [30, 28], inflating 2D convolution kernels to 3D [53] and joint training for visual grounding [59] or captioning [61] are proposed. Very recently, [63] adopt natural language to conduct contrastive training and thus enhance point cloud analysis. Inspired but different from the above, we are the first to conduct image-to-point knowledge distillation for point cloud analysis.

3 Methodlogy

3.1 Problem Statement

Let $\mathcal{P} \in \mathbb{R}^{N\times 3}$ and $y \in \mathbb{R}^1$ be the point cloud and ground-truth label of the 3D object. Its corresponding view-image counterparts can be denoted as $\mathcal{I} \in \mathbb{R}^{V\times H\times W\times 3}$, where N, V and (H, W) are the number of points, number of view-images and image size, respectively. View images can be gained by rendering the 3D CAD model [41] or perspective projecting the raw point cloud [14]. We denote T and S as the image and point cloud analysis networks, respectively, and regard them as the **teacher** and **student** in traditional knowledge distillation (KD). For these networks, we split each

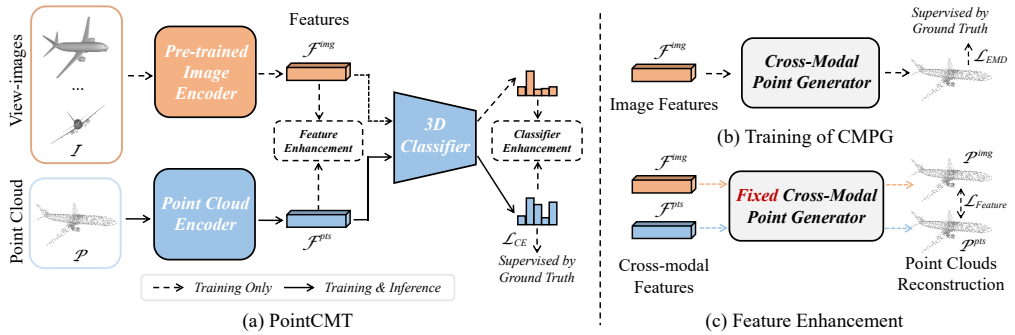

Figure 2: **(a)** The architecture of PointCMT. Multiple view images are gained through rendering the 3D CAD model or perspective projecting the raw point cloud, and the pre-trained image network distills the knowledge to the point cloud analysis network via two matching processes. The first is feature enhancement, which aligns features through a pre-trained cross-modal point generator through the process in (b). The second one is classifier enhancement, which aligns the output distribution of the point classifier by taking cross-modal features as inputs. **(b)** Training process of cross-modal point generator (CMPG). **(c)** Illustration of feature enhancement.

of them into two parts: (i) **Encoders** (*i.e.*, feature extractors $\mathrm{Enc}^{img}(\cdot)$ and $\mathrm{Enc}^{pts}(\cdot)$), the output of which at the last layer are global feature representations $\mathcal{F}^{img} \in \mathbb{R}^D$ and $\mathcal{F}^{pts} \in \mathbb{R}^D$. (ii) **Classifiers**, which project the feature representation $\mathcal{F}^{img}$ and $\mathcal{F}^{pts}$ into class logits through $\mathrm{Cls}^{img}(\mathcal{F}^{img})$ and $\mathrm{Cls}^{pts}(\mathcal{F}^{pts})$, where $\mathrm{Cls}(\cdot)$ denotes the classifier.

Since we formulate the cross-modal learning as a KD problem, its goal in the learning process is to distill the priors knowledge from the image into point cloud features, obtaining an image-enhanced ideal feature $\mathcal{F}^{KD}$. During the knowledge distillation, we parameterize the teacher and student networks with θ_T and θ_S, and denote the knowledge of the teacher network as $\mathcal{K}_T$. From the Bayesian perspective, a neural network can be viewed as a probability model, *e.g.*, $P(y|\mathcal{P}, \theta_S)$ for point cloud analysis model as an example: given an input point cloud $\mathcal{P}$, the network assigns an output probability with the parameters θ_S. Therefore, if we want the student network guided by image knowledge on the input sample, our goal can be further reformulated as maximizing the probability $\mathcal{P}(\mathcal{F}^{KD}|\mathcal{P}, y; \theta_S, \mathcal{K}_T)$, which can also be used to measure the ability of student network extracting the feature with cross-modal information. To find the lower bound of the above probability, we define the discrepancy g between theoretically discriminative features $\mathcal{F}_*^{img}$ and $\mathcal{F}_*^{pts}$ as

$$g = \mathcal{P}(\mathcal{F}_*^{img}|\mathcal{I}, y; \theta_T) - \mathcal{P}(\mathcal{F}_*^{pts}|\mathcal{P}, y; \theta_S, \mathcal{K}_T), \tag{1}$$

where $\mathcal{F}_*^{img}$ and $\mathcal{F}_*^{pts}$ are ideal features in specific modalities. In Lemma 1, we give the lower bound of cross-modal learning as a KD problem. The proof is provided in the supplementary material.

Lemma 1: *By the definition above, $\mathcal{P}(\mathcal{F}^{KD}|\mathcal{P}, y; \theta_S, \mathcal{K}_T)$ is bounded below by $\mathcal{P}(\mathcal{F}^{KD}|\mathcal{I}, y; \theta_T) + \lambda - g$, where λ is*

$$\lambda = \mathcal{P}(\mathcal{F}_*^{img}|\mathcal{I}, y; \theta_T) - \mathcal{P}(\mathcal{F}^{KD}|\mathcal{I}, y; \theta_T). \tag{2}$$

In Lemma 1, λ measures the compatibility of knowledge distillation of the image networks, and can be viewed as a constant when the architectures are determined. $\mathcal{P}(\mathcal{F}^{KD}|\mathcal{I}, y; \theta_T)$ is also a constant when the parameters θ_T and the architecture of the point cloud analysis model are fixed, *e.g.,* using a pre-trained image network. Therefore, the Lemma ensures that during the knowledge distillation, one can maximize $\mathcal{P}(\mathcal{F}^{KD}|\mathcal{P}, y; \theta_S, \mathcal{K}_T)$ through minimizing g, which gives a theoretical guarantee for the KD problem.

If we adopt previous KD studies in cross-modal scenario, they minimize g through making student directly approximate the teacher's features [25, 1, 7] or predict logits [19]. However, in a common KD problem, the teacher and student are generally trained on the same dataset with an identical distribution [19]. Moreover, the teacher generally achieves better performance than the student. In contrast, the image and point cloud analysis models tend to learn different feature representations and logits distribution which are generally complementary. Direct alignment may cause a limited gain or even negative transfer. Moreover, previous KD approaches treat encoders and classifiers as a whole architecture since the teacher and student networks generally have the same components. However, the point cloud convolution is significantly different from the 2D CNN.

Algorithm 1: Process in point cloud cross-modal training (PointCMT)

Data: The point clouds $\{\mathcal{P}_i\}_{i=1}^M$, corresponding ground-truth labels $\{y_i\}_{i=1}^M$ and view-images set $\{\mathcal{I}_i\}_{i=1}^M$, where M is a number of training data samples.

Stage I: Training image encoder and image classifier: Taking view-images set $\{\mathcal{I}_i\}_{i=1}^M$ as input, image encoder produce the image features $\{\mathcal{F}_i^{img}\}_{i=1}^M$, which are then fed into image classifier and obtain prediction logits $\{\text{Cls}^{img}(\mathcal{F}^{img})\}_{i=1}^M$ supervised by ground-truth labels $\{y_i\}_{i=1}^M$.

Stage II: Training cross-modal point generator: As shown in Figure 2 (b), the generator takes features $\{\mathcal{F}_i^{img}\}_{i=1}^M$ as input, reconstructs point cloud $\{\hat{\mathcal{P}}_i^{img}\}_{i=1}^M$ and supervised by point clouds $\{\mathcal{P}_i\}_{i=1}^M$.

Stage III: Image priors assisted training: The point encoder takes point clouds $\{\mathcal{P}_i\}_{i=1}^M$ as input and generate point features $\{\mathcal{F}_i^{pts}\}_{i=1}^M$. The point classifier gains prediction logits $\{\text{Cls}^{pts}(\mathcal{F}^{pts})\}_{i=1}^M$ through taking features $\{\mathcal{F}_i^{pts}\}_{i=1}^M$. The feature enhancement align cross-modal features $\{\mathcal{F}_i^{img}\}_{i=1}^M$ and $\{\mathcal{F}_i^{pts}\}_{i=1}^M$ through **Feature Enhancement**, and **Classifier Enhancement** enhance the point classifier via matching $\{\text{Cls}^{pts}(\mathcal{F}^{pts})\}_{i=1}^M$ and $\{\text{Cls}^{pts}(\mathcal{F}^{img})\}_{i=1}^M$.

To solve the cross-modal KD problem, we propose PointCMT, which effectively solves the cross-modal learning problem. The workflow of PointCMT is demonstrated in Figure 2 (a) and can be summarized as Algorithm 1. Specifically, three stages exist in PointCMT. In Stage I (Section 3.2), we train the image encoder and classifier using view-images and ground-truth labels. In Stage II (Section 3.3), we train the cross-modal point generator (CMPG) through image features, and we independently align features and logits of two modalities in stage III (Section 3.4), where the encoder and classifier of point cloud analysis network are both enhanced.

3.2 Learning Image Priors

For each 3D object, we use multiple view-images (*i.e.*, rendered color images or projected images from raw point cloud) as additional data, and the whole process can be described as:

$$\mathcal{F}^{img} = \mathcal{A}\{\text{CNN}(\mathcal{I}_v)\}_{v=1}^V. \tag{3}$$

Inspired by view-based learning approaches [41], V view-images from $\mathcal{I}$ flow into a shared-weights image feature extractor $\text{CNN}(\cdot)$ to obtain a series of multi-level vectors. By aggregating the view-level vectors via an aggregation function $\mathcal{A}\{\cdot\}$, we obtain a global feature representation $\mathcal{F}^{img}$ from all images, which integrates shape information from multiple views. Finally, an image classifier maps the above global feature to gain a prediction logits through $\text{Cls}^{img}(\mathcal{F}^{img})$, which is supervised by the ground truth label y through cross-entropy loss $\mathcal{L}_{CE}$.

3.3 Cross-Modal Point Generator

A cross-modal point generator (CMPG) can be seen as a nonlinear transformation $\mathbb{R}^D \rightarrow \mathbb{R}^N$ that maps the global feature representation $\mathcal{F}^{img}$ acquired from images into the Euclidean space. Thus, it can avoid potential negative transfer effectively by directly aligning cross-modal features from different distributions. In order to better learn image priors in the point cloud analysis network, we pre-train the CMPG through the image feature $\mathcal{F}^{img}$, and fix it in the distillation stage. Figure 2 (b) illustrates the pre-training stage of CMPG. It takes the image feature as input and reconstruct a point cloud $\hat{\mathcal{P}}^{img}$, which is supervised by the original point cloud $\mathcal{P}$ through Earth Mover's distance (EMD) [39]:

$$\mathcal{L}_{\text{EMD}}(\mathcal{P}, \hat{\mathcal{P}}^{img}) = \min_{\phi} \sum_{p \in \mathcal{P}} ||p - \phi(p)||, \tag{4}$$

where $|\mathcal{P}| = |\mathcal{P}^{img}|$ and $\phi : \mathcal{P} \rightarrow \hat{\mathcal{P}}^{img}$ is a bijection, *i.e.*, for each point $p \in \mathcal{P}$, $\phi(\cdot)$ finds a sole point correspondence in $\hat{\mathcal{P}}^{img}$. In practice, CMPG consists of several fully-connected layers without a specific design. After pre-training, CMPG reconstructs a point cloud through an image-relative feature representation.

3.4 Image Priors Assisted Training

During the training stage, three objectiveness should be optimized:

Classification Loss. In our PointCMT, arbitrary point cloud analysis models can be assembled. Generally, it should be designed through a point encoder and a classifier, where the point encoder takes a point cloud as input, generates the point cloud feature representation $\mathcal{F}^{pts}$ and feeds it into the classifier $\text{Cls}^{pts}(\cdot)$ to obtain a class logits. Finally, the cross-entropy loss $\mathcal{L}_{CE}$ is used as criteria.

Feature Enhancement Loss. Unlike previous KD methods that directly align features of teacher and student, we first transform the cross-modal features into Euclidean space. Specifically, the pre-trained CMPG independently transform $\mathcal{F}^{img}$ and $\mathcal{F}^{pts}$, obtaining two point clouds $\hat{\mathcal{P}}^{img}$ and $\hat{\mathcal{P}}^{pts}$, respectively. After that, EMD loss is conducted on the two-point clouds as an objectiveness:

$$\mathcal{L}_{\text{Feature}} = \mathcal{L}_{\text{EMD}}(\hat{\mathcal{P}}^{pts}, \hat{\mathcal{P}}^{img}) = \min_{\phi} \sum_{p \in \hat{\mathcal{P}}^{pts}} ||p - \phi(p)||, \tag{5}$$

where $|\hat{\mathcal{P}}^{pts}| = |\mathcal{P}^{img}|$ and $\phi : \hat{\mathcal{P}}^{pts} \rightarrow \hat{\mathcal{P}}^{img}$ is a bijection. Compared with traditional L_2 loss, the EMD distance is natural for solving an assignment problem for permutation-invariant point sets. For all but a zero-measure subset of point set pairs, the optimal bijection ϕ is unique and invariant under the infinitesimal movement of the points. Thus, EMD is differentiable almost everywhere.

Classifier Enhancement Loss. In addition to supervising the point feature extractor through the above Feature Enhancement, we further propose constraints conducted on the point classifier. Specifically, the image feature generated by the teacher network is fed into the point classifier, and the output is supervised by the same ground truth, where the gradient only back-propagates to the point classifier. Besides, classifier enhancement is proposed to enable the point classifier to handle the image feature during the distillation. It aligns outputs logits of the point classifier by using image and point features as inputs. This constraint is modified based on the distillation loss in Hinton *et al.* [19] as in Equation (6), but in this case, two sets of logits come from the same classifier.

$$\mathcal{L}_{\text{Hinton}} = \mathcal{D}_{KL}(\text{Cls}^{img}(\mathcal{F}^{img})||\text{Cls}^{pts}(\mathcal{F}^{pts})), \tag{6}$$

where $\mathcal{D}_{KL}(\cdot||\cdot)$ is KL divergence. In contrast, our proposed classifier enhancement is more suitable for the cross-modal scenario where a great discrepancy exist between image and point cloud features. Concretely, the loss for classifier enhancement can be written as

$$\mathcal{L}_{\text{Classifier}} = \mathcal{D}_{KL}(\text{Cls}^{pts}(\mathcal{F}^{img})||\text{Cls}^{pts}(\mathcal{F}^{pts})). \tag{7}$$

Final Loss. The final loss is a weighted sum of the above three losses $\mathcal{L} = \mathcal{L}_{CE} + \alpha\mathcal{L}_{\text{Feature}} + \beta\mathcal{L}_{\text{Classifier}}$, where $\alpha = 30$ and $\beta = 0.3$ are the weights to adjust the ratios of each loss, respectively.

4 Experiment

4.1 Shape Classification on ModelNet40

We firstly evaluate our PointCMT on synthetic dataset ModelNet40 [51], which is a large-scale 3D CAD model dataset.

Dataset description and processing. ModelNet40 is composed of 9,843 train models and 2,468 test models in 40 classes. For the input of the 3D network, we use point clouds provided by the official dataset, which is the same as PointNet [35]. For the input of the image network, we use 20 rendered view images from CAD models utilized in RotationNet [24]. These images have a resolution of 224×224. Since they consider both the mesh surfaces and illumination, they can provide more information to the 3D network. Selected samples of point clouds and corresponding multi-view images are shown in Figure 3.

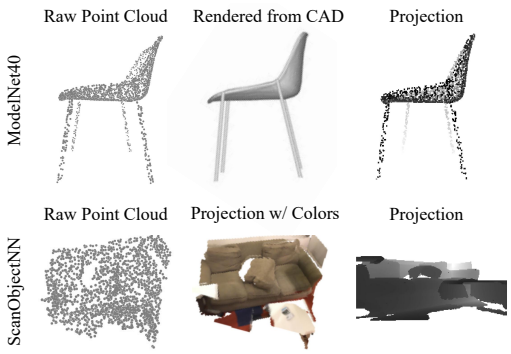

Figure 3: Different view-image generation strategies used in our experiment.

Implementation. For image network, we use ResNet-18 [18] pre-trained on ImageNet [10] as the feature extractor. Following MVCNN [41], we obtain the global shape feature by applying

Table 1: Classification results on ModelNet40 dataset. With only 1k points, PointNet++ trained with PointCMT achieves state-of-the-art results on both class mean accuracy (mAcc) and overall accuracy (OA) metrics. Here, 'pnt' and 'nor' denote points and normal vectors, respectively. The speed (samples/second) tested on one Tesla V100 GPU and four cores AMD EPYC 7351@2.60GHz CPUs, where † denotes the results from the original paper. * For PointNet++, we train it with protocol of RS-CNN [29] as mentioned in [14]. The best and second best are marked in **bold** and underline.

Method	Input	#Points	mAcc(%)	OA(%)	Speed	Param.
PointNet [35]	pnt	1k	86.0	89.2	-	3.47M
PointNet++ [36]	pnt, nor	5k	-	91.9	-	1.47M
PointCNN [27]	pnt	1k	88.0	92.5	-	-
PointConv [50]	pnt, nor	1k	-	92.5	80†	18.6M
KPConv [42]	pnt	7k	-	92.9	10†	15.2M
PointASNL [56]	pnt, nor	1k	-	93.2	-	-
PosPool [31]	pnt	5k	-	93.2	-	-
Point Transformer [64]	pnt	1k	90.6	93.7	-	-
GBNet [38]	pnt	1k	91.0	93.8	112†	8.4M
GDANet [54]	pnt	1k	-	93.8	14†	0.9M
SimpleView [14]	pnt	1k	-	93.9	2208	1.64M
CurveNet [52]	pnt	1k	-	94.2	15†	2.0M
PointMLP [33]	pnt	1k	**91.4**	**94.5**	139	12.6M
DGCNN [46] (baseline)	pnt	1k	90.2	92.9	518	1.68M
RS-CNN [29] (baseline)	pnt	1k	89.3	92.9	2174	1.17M
PointNet++ [36] (baseline)	pnt	1k	90.1	93.4*	300	1.62M
DGCNN **w/** PointCMT	pnt	1k	90.8 (+0.6)	93.5 (+0.6)	518	1.69M
RS-CNN **w/** PointCMT	pnt	1k	90.1 (+0.8)	93.8 (+0.9)	2174	1.17M
PointNet++ **w/** PointCMT	pnt	1k	91.2 (+1.1)	94.4 (+1.0)	300	1.62M

view-wise max-pooling to the view-level features. Finally, a fully-connected layer is used to output the classification logits. During the training process of the above network, we use SGD as our optimizer with a learning rate of 0.01. The batch size is set to 128 for 50 epochs. After that, we fix the image network and train the CMPG with Adam and 32 batch size for 50 epochs. In practice, CMPG consists of three-layer MLP. For the point cloud analysis models, DGCNN [46] and RS-CNN [29] are independently trained with the training strategies provided in their official codes. PointNet++ [36] is trained with strategy of RS-CNN [29] as [14], since it can gains higher performance.

Comparison with state-of-the-arts. The classification results on ModelNet40 are shown in Table 1, where the overall accuracy (OA) and class mean accuracy (mAcc) are compared. The upper part of the table illustrates the results of current state-of-the-art methods, in which we use PointNet++ [36], RS-CNN [29] and DGCNN [46] as our baselines. **We do not use PointMLP [33] as our baseline since it cannot robustly reproduce the highest results on ModelNet40, where the issue is mentioned in their open-sourced codes.** For models trained from scratch, PointMLP [33] achieves the highest accuracy. As shown in the lower part of the table, after training with PointCMT, the performance of all baselines is greatly boosted, *i.e.,* 1.0% improvement upon PointNet++ and 0.9% for RS-CNN and 0.6% for DGCNN. We also compare our method to several open-sourced methods and report the parameters and testing speed. As shown in the last two columns of the table, though PointMLP gains 0.1 higher overall accuracy, its network consists of about 7.7× parameters and only achieves 46% speed of PointNet++. In contrast, PointCMT performs well on light-weighted models, which shows its great potential for real-time applications, *e.g.*, scene parsing in autonomous driving.

4.2 Shape Classification on ScanObjectNN

Though ModelNet40 is the widely used benchmark for point cloud analysis, it may not meet the realistic requirement due to its synthetic nature. To this end, we also conduct experiments on the ScanObjectNN benchmark [44], which is a real-world dataset.

Dataset description and processing. ScanObjectNN collects 2,902 objects from real-world indoor scenes ScanNet [9] and SceneNN [21], categorizing into 15 categories. Several variants are provided in the dataset, where the most challenging one is PB_T50_RS, *i.e.*, introducing perturbation objects (11,416 and 2,882 data for training and test) via random translation, shift, rotation and scaling. Due to background, noise, and occlusions, this benchmark poses significant challenges to existing point cloud

Table 2: Classification on ScanObjectNN. We examine all methods on original objects (OBJ_ONLY) and more challenging variant (PB_T50_RS). * We train PointNet++ and PointMLP with protocol of [14] for a fair comparison. The best and second best are marked in **bold** and underline.

Method	OBJ_ONLY		PB_T50_RS	
	mAcc(%)	OA(%)	mAcc(%)	OA(%)
3DmFV [4]	-	73.8	58.1	63.0
PointNet [35]	-	79.2	63.4	68.2
SpiderCNN [55]	-	79.5	69.8	73.7
PointNet++ [36]	-	84.3	75.4	77.9
DGCNN [46]	-	86.2	73.6	78.1
PointCNN [27]	-	85.5	75.1	78.5
DRNet [37]	-	-	78.0	80.3
GBNet [38]	-	-	77.8	80.5
SimpleView [14]	86.2	89.0	-	80.8
PRANet [6]	-	-	79.1	82.1
MVTN [17]	-	-	-	82.8
PointNet++ [36] (baseline)	85.6	87.5	75.8*	79.4*
PointMLP [33] (baseline)	<u>89.4</u>	<u>92.5</u>	<u>84.3</u>	<u>85.7</u>
PointNet++ **w/** PointCMT	89.3 (+3.7)	91.8 (+4.3)	80.2 (+4.4)	83.3 (+3.9)
PointMLP **w/** PointCMT	**92.0** (+2.6)	**93.5** (+1.0)	**84.8** (+0.5)	**86.7** (+1.0)

analysis methods. Furthermore, since the PB_T50_RS dataset only preserves the spatial coordinates (XYZ) for each object while the other information, such as RGB, is discarded, we also compared the original 2,902 objects (OBJ_ONLY), which includes additional RGB information. On both above datasets, we only use depth images through conducting perspective projection on raw point cloud as additional inputs, as shown in the last column in Figure 3. Section 4.6 will discuss more results using projection with additional color information.

Implementation. All view-image in ScanObjectNN are gained by the projection of raw point clouds, and we follow the structure of [14] only generate six images for PB_T50_RS and OBJ_ONLY. We train the image network from scratch with batch size 32 and SGD optimizer for longer epochs of 1,000. The training strategy of CMPG is the same as ModelNet40. For both point cloud models trained from scratch and with PointCMT, we use SGD optimizer for 1,000 epochs with a batch size of 32 as [33].

Comparison with state-of-the-arts. The results are shown in Table 2, where PointNet++ [36] and current state-of-the-art PointMLP [33] are chosen as our baselines. PointCMT significantly improves the performance on both class mean accuracy (mAcc) and the overall accuracy (OA), even on state-of-the-art methods. Specifically, although background, noise, and occlusions exist on PB_T50_RS dataset, PointCMT still improves the overall accuracy of PointNet++ by 3.9%. Moreover, PointCMT also achieves state-of-the-art results on OBJ_ONLY dataset. Note that there is no auxiliary information provided in images, and all view images are generated through the perspective projection of points coordinates. Nevertheless, PointCMT still dramatically increases the mAcc of PointMLP from 89.4% to 92.0% (+2.6%).

4.3 Data Efficient Learning

We evaluate our approach under limited data scenarios in Table 3. Here, we only sample a small amount of training data in each category on ModelNet40, and evaluate the entire testing data. Our PointCMT shows an even more significant gap when using a small subset of the training data, again compared to PointNet++ which trained from scratch. Especially when facing only 2% and 10% of the training data, we achieve about 1.9% and 2.8% improvements, respectively. This result illustrates that PointCMT provides more vital guidance for point cloud models in the low data regime.

Table 3: Data efficient learning on ModelNet40. We train PointNet++ [36] with a small amount of training data and train with PointCMT.

Data percentage	Train from scratch	w/ PointCMT
2%	73.3	**75.2** (+1.9)
5%	82.1	**83.5** (+1.4)
10%	85.1	**87.9** (+2.8)
20%	88.4	**89.3** (+0.9)

Table 4: Ablation study on ModelNet40 and ScanObjetNN datasets. Overall accuracy (%) as metrics.

Model	FE	CE	ModelNet40	OBJ_ONLY	PB_T50_RS
PointNet++	✗	✗	93.4	87.5	79.4
	✓	✗	93.8 (+0.4)	89.2 (+1.7)	82.5 (+3.1)
	✗	✓	94.0 (+0.6)	91.3 (+3.8)	82.3 (+2.9)
	✓	✓	94.4 (+1.0)	91.8 (+4.3)	83.3 (+3.9)

4.4 Ablation study

The ablation results on three datasets are summarized in Table 4, in which we use PointNet++ as our baselines. We first test the effectiveness of feature enhancement (FE) and classifier enhancement (CE) in PointCMT. The results demonstrate that only using FE already significantly boosts the performance on both datasets, *i.e.,* increases the overall accuracy by 0.4% and 3.1% on ModelNet40 and ScanObjetNN. Only using classifier enhancement (CE) improves the accuracy by around 0.6% and 2.9%. Finally, it is surprising that when we use both FE and CE during the training phase, it achieves the best result of 94.4% and 83.3%, respectively.

4.5 Comparison with Knowledge Distillation Methods

To further verify the effectiveness of our proposed method compared with typical teach-student architecture and other knowledge distillation manners, we compare PointCMT with typical approaches of knowledge transfer in Table 5. Among all the methods, Hinton *et al.* [19] is a pioneer study for knowledge distillation, while Huang *et al.* [23] and Yang *et al.* [57] are recent works. As shown in the table, directly aligning features as [19] between two modalities will cause a negative transfer on ModelNet40. This phenomenon does not appear on ScanObjectNN, since view images projected via point clouds may have a smaller gap than the rendered CAD model. Nevertheless, other KD techniques only achieve marginal improvement compared with PointCMT.

Table 5: Comparison with knowledge distillation methods. We compare overall accuracy (OA,%) gained by PointNet++ on ModelNet40 and ScanObjectNN.

Method	ModelNet40	PB_T50_RS
Baseline	93.4	79.4
Hinton *et al.* [19]	93.1 (-0.3)	81.8 (+2.4)
Huang *et al.* [23]	93.6 (+0.2)	82.0 (+2.6)
Yang *et al.* [57]	93.7 (+0.3)	81.1 (+1.7)
PointCMT (ours)	**94.4** (+1.0)	**83.3** (+3.9)

4.6 Different View-image Generation

In this section, we compare results with different view-image generation strategies. As shown in Figure. 3, multiple view-image types can be applied in our framework, and we compare the results in Table 6. As illustrated in the table, images rendered from a CAD model improve the result compared with only using projection since it provides additional shade and texture information. In contrast, we find that using additional colors in the OBJ_ONLY dataset cannot boost performance. The reason is that OBJ_ONLY dataset only contains 2,902 objects, and image networks are easier to overfit when using the color information.

Table 6: Results through different view-image generation. We compare overall accuracy (OA,%) on ModelNet40 and ScanObjectNN OBJ_ONLY datasets.

View-images	ModelNet40	OBJ_ONLY
Rendered from CAD	**94.4**	-
Projection	94.0	**91.8**
Projection w/ color	-	90.7

5 Conclusion

In this work, we propose a point cloud cross-modal training strategy named PointCMT. By exploiting some sophisticated architecture and reasonable criteria function, our PointCMT can boost the performance significantly for point cloud analysis methods on several benchmarks, outperforming previous methods by a large margin. We believe that our work can be applied to a broader range of other scenarios in the future, such as 3D semantic segmentation and object detection. Meanwhile, our method provides an alternative solution to the comprehension of 3D scenes with severe texture details missing. It can improve performance through image priors and knowledge transfer.

References

[1] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9163–9171, 2019.

[2] Lei Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? *NeurIPS*, page 2654–2662, 2014.

[3] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9297–9307, 2019.

[4] Yizhak Ben-Shabat, Michael Lindenbaum, and Anath Fischer. 3dmfv: Three-dimensional point cloud classification in real-time using convolutional neural networks. *IEEE Robotics and Automation Letters*, 3(4):3145–3152, 2018.

[5] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 742–751, 2017.

[6] Silin Cheng, Xiwu Chen, Xinwei He, Zhe Liu, and Xiang Bai. Pra-net: Point relation-aware network for 3d point cloud analysis. *IEEE Transactions on Image Processing*, 30:4436–4448, 2021.

[7] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4794–4802, 2019.

[8] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019.

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

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[11] Xiang Deng and Zhongfei Zhang. Comprehensive knowledge distillation with causal intervention. *Advances in Neural Information Processing Systems*, 34, 2021.

[12] Khaled El Madawi, Hazem Rashed, Ahmad El Sallab, Omar Nasr, Hanan Kamel, and Senthil Yogamani. Rgb and lidar fusion based 3d semantic segmentation for autonomous driving. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 7–12. IEEE, 2019.

[13] Yifan Feng, Zizhao Zhang, Xibin Zhao, Rongrong Ji, and Yue Gao. Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, June 2018.

[14] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. In *International Conference on Machine Learning*, pages 3809–3820. PMLR, 2021.

[15] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9224–9232, 2018.

[16] 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, 2021.

[17] Abdullah Hamdi, Silvio Giancola, and Bernard Ghanem. Mvtn: Multi-view transformation network for 3d shape recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–11, 2021.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *NeurIPS Workshops*, 2014.

[20] Ji Hou, Saining Xie, Benjamin Graham, Angela Dai, and Matthias Nießner. Pri3d: Can 3d priors help 2d representation learning? *arXiv preprint arXiv:2104.11225*, 2021.

[21] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 92–101. IEEE, 2016.

[22] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 984–993, 2018.

[23] Zhen Huang, Xu Shen, Jun Xing, Tongliang Liu, Xinmei Tian, Houqiang Li, Bing Deng, Jianqiang Huang, and Xian-Sheng Hua. Revisiting knowledge distillation: An inheritance and exploration framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3579–3588, 2021.

[24] 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*, pages 5010–5019, 2018.

[25] Jangho Kim, SeongUk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. *Advances in neural information processing systems*, 31, 2018.

[26] Georg Krispel, Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Fuseseg: Lidar point cloud segmentation fusing multi-modal data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1874–1883, 2020.

[27] 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*, 31, 2018.

[28] Zhenyu Li, Zehui Chen, Ang Li, Liangji Fang, Qinhong Jiang, Xianming Liu, Junjun Jiang, Bolei Zhou, and Hang Zhao. Simipu: Simple 2d image and 3d point cloud unsupervised pre-training for spatial-aware visual representations. *arXiv preprint arXiv:2112.04680*, 2021.

[29] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019.

[30] Yueh-Cheng Liu, Yu-Kai Huang, Hung-Yueh Chiang, Hung-Ting Su, Zhe-Yu Liu, Chin-Tang Chen, Ching-Yu Tseng, and Winston H Hsu. Learning from 2d: Pixel-to-point knowledge transfer for 3d pretraining. *arXiv preprint arXiv:2104.04687*, 2021.

[31] Ze Liu, Han Hu, Yue Cao, Zheng Zhang, and Xin Tong. A closer look at local aggregation operators in point cloud analysis. In *European Conference on Computer Vision*, pages 326–342. Springer, 2020.

[32] Zhengzhe Liu, Xiaojuan Qi, and Chi-Wing Fu. 3d-to-2d distillation for indoor scene parsing. *CVPR*, 2021.

[33] 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*, 2022.

[34] Gregory P Meyer, Jake Charland, Darshan Hegde, Ankit Laddha, and Carlos Vallespi-Gonzalez. Sensor fusion for joint 3d object detection and semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[35] Charles R Qi, Hao Su, Kaichun Mo, and LeonidasJ Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[36] 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*, pages 5099–5108, 2017.

[37] Shi Qiu, Saeed Anwar, and Nick Barnes. Dense-resolution network for point cloud classification and segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3813–3822, 2021.

[38] Shi Qiu, Saeed Anwar, and Nick Barnes. Geometric back-projection network for point cloud classification. *IEEE Transactions on Multimedia*, 2021.

[39] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

[40] Suraj Srinivas and François Fleuret. Knowledge transfer with jacobian matching. In *International Conference on Machine Learning*, pages 4723–4731. PMLR, 2018.

[41] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. ICCV*, 2015.

[42] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, Francois Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[43] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

[44] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1588–1597, 2019.

[45] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4604–4612, 2020.

[46] 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.

[47] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38(5):1–12, 2019.

[48] 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*, pages 1850–1859, 2020.

[49] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019.

[50] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019.

[51] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[52] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 915–924, 2021.

[53] Chenfeng Xu, Shijia Yang, Bohan Zhai, Bichen Wu, Xiangyu Yue, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Image2point: 3d point-cloud understanding with pretrained 2d convnets. *arXiv preprint arXiv:2106.04180*, 2021.

[54] Mutian Xu, Junhao Zhang, Zhipeng Zhou, Mingye Xu, Xiaojuan Qi, and Yu Qiao. Learning geometry-disentangled representation for complementary understanding of 3d object point cloud. *arXiv preprint arXiv:2012.10921*, 2, 2021.

[55] 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)*, pages 87–102, 2018.

[56] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5589–5598, 2020.

[57] Jing Yang, Brais Martinez, Adrian Bulat, and Georgios Tzimiropoulos. Knowledge distillation via softmax regression representation learning. In *International Conference on Learning Representations*, 2020.

[58] Ze Yang and Liwei Wang. Learning relationships for multi-view 3d object recognition. In *Proc. IEEE Int. Conf. Comput. Vision*, pages 7505–7514, 2019.

[59] Zhengyuan Yang, Songyang Zhang, Liwei Wang, and Jiebo Luo. Sat: 2d semantics assisted training for 3d visual grounding. 2021.

[60] Haoxuan You, Yifan Feng, Rongrong Ji, and Yue Gao. Pvnet: A joint convolutional network of point cloud and multi-view for 3d shape recognition. In *Proc. ACM Int. Conf. Multimedia*, MM '18, pages 1310–1318, New York, NY, USA, 2018. ACM.

[61] Zhihao Yuan, Xu Yan, Yinghong Liao, Yao Guo, Guanbin Li, Zhen Li, and Shuguang Cui. X-trans2cap: Cross-modal knowledge transfer using transformer for 3d dense captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[62] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *ICLR*, 2017.

[63] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. *arXiv preprint arXiv:2112.02413*, 2021.

[64] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021.

[65] Helong Zhou, Liangchen Song, Jiajie Chen, Ye Zhou, Guoli Wang, Junsong Yuan, and Qian Zhang. Rethinking soft labels for knowledge distillation: A bias-variance tradeoff perspective. *arXiv preprint arXiv:2102.00650*, 2021.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes]
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes]

   (b) Did you mention the license of the assets? [No]

   (c) Did you include any new assets either in the supplemental material or as a URL? [No]

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]