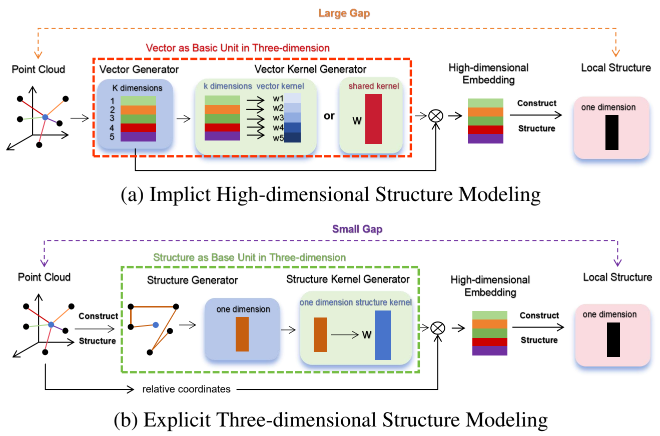
X-3D: Explicit 3D Structure Modeling for Point Cloud Recognition

Shuofeng Sun[1], Yongming Rao[2], Jiwen Lu[3], Haibin Yan[1]*
Beijing University of Posts and Telecommunications[1], Tencent[2], Tsinghua University[3]

Abstract

*Numerous prior studies predominantly emphasize constructing relation vectors for individual neighborhood points and generating dynamic kernels for each vector and embedding these into high-dimensional spaces to capture implicit local structures. However, we contend that such implicit high-dimensional structure modeling approch inadequately represents the local geometric structure of point clouds due to the absence of explicit structural information. Hence, we introduce X-3D, an explicit 3D structure modeling approach. X-3D functions by capturing the explicit local structural information within the input 3D space and employing it to produce dynamic kernels with shared weights for all neighborhood points within the current local region. This modeling approach introduces effective geometric prior and significantly diminishes the disparity between the local structure of the embedding space and the original input point cloud, thereby improving the extraction of local features. Experiments show that our method can be used on a variety of methods and achieves state-of-the-art performance on segmentation, classification, detection tasks with lower extra computational cost, such as **90.7%** on ScanObjectNN for classification, **79.2%** on S3DIS 6 fold and **74.3%** on S3DIS Area 5 for segmentation, **76.3%** on ScanNetV2 for segmentation and **64.5%** mAP$_{25}$, **46.9%** mAP$_{50}$ on SUN RGB-D and **69.0%** mAP$_{25}$, **51.1%** mAP$_{50}$ on ScanNetV2. Our code is available at https://github.com/sunshuofeng/X-3D.*



(a) Implict High-dimensional Structure Modeling

(b) Explicit Three-dimensional Structure Modeling

Figure 1. Illustration of the different design paradigms. **Implict High-dimensional Structure Modeling (IHSM)**. Most of existing work can be classified into this design paradigm, and the focus of modeling usually lies in how to construct relation vectors for each neighborhood point or how to generate a dynamic kernel for each vector and then embed the relation vectors to the high-dimension space to capture the implicit local structure. **Explicit 3D Structure Modeling**. The difference is that the basic unit of our modeling is the structure, and we directly build the explicit geometric structure for the local neighborhood in the input space, and generate the dynamic kernel which shares weights for all neighborhood points within the current local region through the geometric structure. By this explicit introduction of structural information into the embedding space, we greatly reduce the gap between the local structure captured by the embedding space and the original input point cloud.

1. Introduction

Using deep learning for point cloud analysis has garnered significant attention in research. However, the irregularity of point cloud data poses challenges for directly applying conventional convolutional methods.

In order to solve this problem, the existing methods [11, 17, 18, 24, 27, 32, 43] first construct the local neighborhood by kNN or ball query, and then design the relationship vector to represent the geometric relationship between each neighborhood point and the center point. For example, PointNet++ [24], PointNeXt [27], and other methods directly use the relative coordinate difference as the relationship vector. Methods such as RandLA [12] and RSConv [18] supplement the relation vectors with additional Euclidean distances and absolute coordinates. Then most methods use shared-weight MLP to embed the relation vector into the high-dimensional feature space. However, in order to flexibly handle different relation vectors, some methods [18, 32] dynamically generate the corresponding vector kernel for each relation vector. Finally, the implicit local structure is obtained by applying a symmetric aggre-

*Corresponding author

gation function like attention mechanism [14, 43] or max-pooling to all embedding vectors. We refer to the above modeling process as implicit high-dimensional structure modeling, where the basic units are vectors, and then the local structure is only implicitly captured in high-dimensional space.

However, there may be some problems with the above paradigm. First, local structure is implicitly captured in high dimensions which may lead to the loss of some detailed information and a large gap with the local structure of the original data. Further, point clouds are typical manifold data in the non-Euclidean space and thus relations vectors in the Euclidean space may provide inaccurate geometric information (e.g., Euclidean distances are very close, and geodesic distances are very far). To address this issue, we propose X-3D, an explicit 3D structure modeling paradigm, which is shown in Figure 1. X-3D directly constructs and represents the local structure in the original 3D input space, generating structure kernels dynamically which is shown in Figure 2. Each relation vector is embedded using a dynamically generated kernel specific to its local region. As a result, the output embedding vector is highly correlated with the 3D local structure it represents. This significantly reduces the gap between the embedding space and the original input space's local structure, enabling more effective extraction of local features

Furthermore, to improve X-3D's performance, we focus on improving the neighborhood quality through denoising and neighborhood context propagation. Initially, denoising involves removing outliers from the explicit structure by employing a cross-attention mechanism between neighborhood points and the explicit local structure. Subsequently, to refine the neighborhood representation, we dynamically propagate neighborhood context information. However, to avoid conflating distant geometric structures and conflicting with local information, the scope of neighborhood context propagation is restricted.

Our contributions can be summarized as follows:

- We propose an explicit 3D structure modeling approach.
- We propose denoising and neighborhood context propagation to further improve the performance.
- We conduct an in-depth analysis from the viewpoint of manifold learning.
- Our method can be embedded into other models and reach state-of-the-art performance.

2. Related Work

Voxel-based Methods: Due to the irregularity of the point cloud, many works [4, 7, 36, 37, 44] voxelize the input point cloud, transform it into a regular grid, and use 3D convolution for processing. However it is easy to miss the detailed information of the original point cloud.

Point-based Methods: Some researchers perform feature extraction directly on the raw point cloud. For example, PointNet++ [24] constructed a neighborhood, then extracts each neighborhood point feature, and finally captures the implicit local structure of the neighborhood through a symmetric aggregation function. The subsequent work [11, 18, 28] designed a rich relationship vector to describe the neighborhood point features based on the PointNet++ design paradigm, and graph-based work [16, 30, 34] designed different edge descriptors to extract the neighborhood features, and dynamic weight (kernel)-based [12, 18, 32] dynamically generated the corresponding kernel or weight for each neighborhood point.

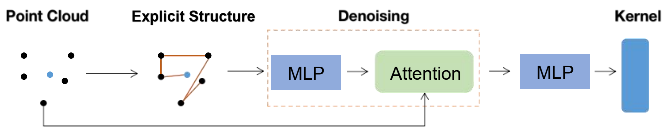
Manifold Learning: Manifold data lies in the non-Euclidean space and has complex local structure. Simple Euclidean metric is difficult to measure the geometric properties of manifold data and easily leads to capture the wrong geometric structure. ISOMAP captures the manifold properties by computing geodesic distances instead of Euclidean metrics and minimizing point distances across all input and embedding spaces, while methods such as LLE [29], LTSA [41] captures the correct manifold surface by minimizing the local structure of the original input and embedding spaces. Point clouds are typical manifold data, and simply using relative coordinates to represent the geometry is not enough. Therefore, we propose to explicitly exploit the local structure of the input space.

3. Background

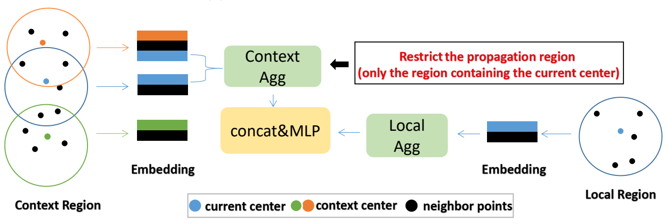
In this part, we introduce implict high-dimensional structure modeling (IHSM) in detail. Most existing methods [11, 12, 17, 18, 24, 28, 32] can be considered as this paradigm, where there are two main key designs. The first key design is how to design the relation vectors in the original input space, and the second key design is how to design the corresponding dynamic kernel for each relation vector and embed each vector to the high-dimension space. It should be noted that the shared-weight MLP can be considered as a special case of the second one.

A point cloud can be represented as a set of N points, we use $\mathbf{p} \in R^{N\times3}$ to represent the coordinates of the points and $\mathbf{f} \in R^{N\times C}$ represents the features of the points, where C is the feature dimension of the points. For the i-th center point $\mathbf{p}_i$, we define its neighborhood points as $\mathbf{p}_{i,j}$, where $j = 1, 2...k$, k is the neighborhood number, and then the relation vector between the neighborhood points and the center point can be written as $V_{i,j}$.

Many works focus on how to extract local structures by constructing appropriate relation vectors, for example, PointNet++ [24], PointMetaBase [17] and others simply treat relative coordinate differences as relation vectors, while RandLA [12] adds absolute coordinates and Euclidean distances. GAM [11] and RepSurf [28] consider



(a) Structure Kernel Generator



(b) Neighborhood Context Propagation

Figure 2. Illustration of X-3D. **(a)** first constructs the explicit local structure from the original input space, then reduces the influence of noise points on the local structure by cross attention, and finally generates the structure kernel by MLP. **(b)** avoids the influence caused by random neighborhood selection by propagating the neighborhood context,. Furthermore, by limiting the scope of dynamic context propagation, it ensures that the explicit local structure does not conflict

depth gradient information and curvature, respectively. In particular, RepSurf explicitly incorporates local structure during curvature construction, primarily performed during preprocessing. However, as the down-sampling rate and local neighborhood half-size increase, RepSurf struggles to accurately portray the region's local structure. Consequently, it tends to describe solely point and vector information, leading to its attribution to IHSM.

Then for each vector, some works generate its corresponding dynamic vector kernel, which can be written as:

$$\mathbf{W}_{i,j} = \theta(V_{i,j}) \tag{1}$$

For example, RSConv [18] simply uses MLP to generate dynamic kernels based on relation vectors and KPConv [32] dynamically generates vector kernels by calculating the geometric relationship between each neighborhood point and the predefined kernel points. In particular, for shared-weight MLP, it can be rewritten as:

$$\mathbf{W}_{i,j} = \theta \tag{2}$$

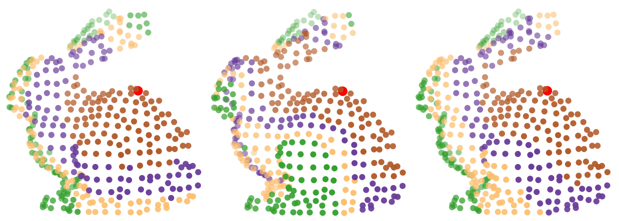
Then, the process the vector modeling can be written as

$$\hat{\mathbf{f}}_{i,j} = \mathcal{M}(\mathbf{W}_{i,j}, \mathbf{f}_{i,j}, V_{i,j}) \tag{3}$$

where $\mathcal{M}$ represents how to update neighborhood features according to dynamic vector kernel and relation vector.

Finally, the implicit local structure is obtained by aggregating the modeled relation vectors in the neighborhood, which can be written as:

$$LS_i = \underset{j=1..k}{Agg}\,(\hat{\mathbf{f}}_{i,j}) = \underset{j=1..k}{Agg}\,(\mathcal{M}(\theta(V_{i,j}), \mathbf{f}_{i,j}, V_{i,j})) \tag{4}$$



(a) Geodesic Distance (b) IHSM (c) Ours

Figure 3. Examples of local structures captured by different methods. **(a)** captures local structure by computing geodesic distances on the original input point cloud. **(b)** captures the implicit local structure through implicit high-dimensional structure modeling, and it can be seen that there is still a certain difference from the local structure of the original space represented by the geodesic distance. **(c)** captures the local structure through X-3D, and it can be seen that the local structure difference from the original space represented by the geodesic distance is small, and the representation in the edge part is more reasonable.

We can see that the basic unit that captures implicit local structure is the vector. Specifically, both the input and the corresponding kernel only consider the relation vector and ignore the overall structure. Therefore, many details are missed in the transformation from vector to structure in high-dimensional space, which may lead to a large gap with the local structure in the input point cloud data and increase the difficulty of model learning (An example in Figure 3).

4. Proposed Method

To solve the problem of IHSM, we change the focus of design from vectors to structures and propose explicit 3D structure modeling. In simple terms, we capture explicit local structure directly in the raw input point cloud and dynamically generate structure kernels based on it, which can be written as:

$$\hat{\mathbf{W}}_i = \theta(ES_i) \tag{5}$$

where ES_i represents the explicit structure of the neighborhood to the i-th center point. Then the neighborhood point features are updated by the following formula:

$$\hat{\mathbf{f}}_{i,j} = \mathcal{M}(\hat{\mathbf{W}}_i, \mathbf{f}_{i,j}, \mathbf{p}_i - \mathbf{p}_{i,j}) \tag{6}$$

Finally, the capture of local structure in the embedding space can be written as:

$$LS_i = \underset{j=1..k}{Agg}\,(\hat{\mathbf{f}}_{i,j}) = \underset{j=1..k}{Agg}\,(\mathcal{M}(\theta(ES_i), \mathbf{f}_{i,j}, \mathbf{p}_i - \mathbf{p}_{i,j})) \tag{7}$$

We can see that the similarity of (4) and (7) effectively indicates the applicability of X-3D, so the proposed approach can be easily embedded into most of the existing models and effectively improve their performance.

In the following part, we will go through the various parts of X-3D in detail.

4.1. Explict Structure

For the explicit structure ES_i, explicitly capturing local structure has been deeply studied in previous machine learning methods, such as LLE [29], which takes the linear reconstruction relationship **(LR)** between the center point and the neighborhood points as the local structure, PointHop [40], which divides the point cloud neighborhood into fixed octuples and extracts the centroid coordinates of each partition **(PH)**, and some methods [6, 15] utilize neighborhood **PCA** to combine eigenvector and eigenvalue to obtain robust structural descriptors.

We conducted a large number of experiments to explore which ES_i has better performance. Through extensive experiments, we found that effective explicit local structures ES_i should have the following properties:

- **Symmetry:** For a neighborhood of k points, the 1D ES_i should be symmetric, and the order of the points does not affect the capture of the explicit structure.
- **Global Shape:** ES_i should be able to describe the global shape of the local region, which provides a good geometric prior for tasks such as segmentation.
- **Local Details:** ES_i should simply describe the local details of the current local region, which contributes to the correct geometric correlation of the points in the current neighborhood.
- **Explicit information:** ES_i should be explicitly derived from the data rather than implicitly learned by the model

In Table 1, we evaluated different explicit local structures to underscore the significance of these properties. Results indicate that **IS** exhibits lower performance on GAP and OA due to a lack of explicit structural information. Similarly, **LR** lacks symmetry, affecting its generalization performance, particularly in the presence of disorder within the point cloud, resulting in lower overall accuracy (OA). Considering the higher performance and comprehensive properties of PointHop, we default to using it as the explicit local structure. The process of PointHop can be written as:

$$\overline{\mathbf{p}}_{i,c} = \frac{1}{K_{i,c}} \sum_{j=1}^{k} t_{i,j,c} \mathbf{p}_{i,j}, c = 1, ...8 \tag{8}$$

$$ES_i = Concat([\overline{\mathbf{p}}_{i,1}, ...\overline{\mathbf{p}}_{i,8}]) \tag{9}$$

where $\overline{\mathbf{p}}_{i,c}$ represents the centroid coordinates of the c-th partition of the i-th point, and $K_{i,c}$ represents the number of neighborhood points of the partition. $t_{i,j,c}$ represents whether the j-th neighborhood point of the i-th center point belongs to the c-th partition,which can be written as:

$$t_{i,j,c} = \begin{cases} 1, & \mathbf{p}_{i,j} \in \xi_c \\ 0, & \mathbf{p}_{i,j} \notin \xi_c \end{cases} \tag{10}$$

Table 1. We test the effect of different local structures on ScanObjectNN and report **OA** (overall accuracy). We additionally report the Euclidean distance (**GAP**) between the local structure of the embedding space and the local structure of the input space. To test the need for explicit structure, we add ablation experiments with implicit structure (**IS**) learned through the network.

	Symmetry	Glocal Shape	explicit	Local Details	GAP	OA
LR			✓	✓	1.48	87.2
IS	✓	✓		✓	5.37	87.6
PH	✓	✓	✓	✓	**1.36**	**88.8**
PCA	✓	✓	✓		1.53	88.6

where ξ_c represents the c-th partition.

4.2. Structure Kernel Generator.

Having obtained the explicit local structure ES_i, we use it to dynamically generate structure kernels. First, we further extract local structural information by:

$$F_i = MLP(ES_i) \tag{11}$$

Then we simply define noisy points as those that "differ significantly from the local structure", therefore we calculate the correlation between each point and local structure information, and perform adaptive fusion to eliminate the influence of noise points:

$$s_{i,j} = Softmax(MLP(\mathbf{p}_{i,j})^T F_i) \tag{12}$$

$$F_i = F_i + \sum_{j=1}^{K} s_{i,j} MLP(\mathbf{p}_{i,j}) \tag{13}$$

Then we expect the model to learn the intrinsic geometric properties of the point cloud surface through the local geometric structure, capturing the correct geometric correlations. To do this, we use the geometric structure descriptor to dynamically generate kernels.

$$\hat{\mathbf{W}}_i = MLP(F_i) \tag{14}$$

Then, in the process of updating the neighborhood point features, based on the practice of PointMetaBase, we explicitly add the dynamic position embedding to the neighborhood point features to provide the correct geometric correlation.

$$\mathcal{M}(\hat{\mathbf{W}}_i, \mathbf{f}_{i,j}, \mathbf{p}_i - \mathbf{p}_{i,j}) = \hat{\mathbf{W}}_i(\mathbf{p}_i - \mathbf{p}_{i,j}) + MLP(\mathbf{f}_{i,j}) \tag{15}$$

4.3. Neighborhood Context Propagation.

In order to reduce the influence caused by random neighborhood point selection, we dynamically propagate

Table 2. Semantic segmentation results on the S3DIS [1] 6-Fold and Area-5 dataset and the ScanNetV2 [5] validation set. Our X-3D can effectively improve its performance on the state-of-the-art large models as well as the small-scale models, and reach the new state-of-the-art (**rank1 without extra training data**) with only a small computational overhead.

Method	**S3DIS 6-fold**			**S3DIS Area-5**			**ScanNet**	GFLOPs	#Params
	mIoU	mAcc	OA	mIoU	mAcc	OA	mIoU		
KPConv [32]	70.6	79.1	-	67.1	72.8	-	68.4	-	-
PointTransformer [43]	73.5	81.9	90.2	70.4	76.5	90.8	70.6	2.80	-
RepSurf [28]	74.3	82.6	90.8	68.9	76.0	90.2	70.0	1.04	-
RandLA [12]	70.0	81.5	88.0	-	-	-	-	5.8	-
PTV2 [35]	-	-	-	71.6	77.9	91.1	75.4	-	-
PointNeXt-XL [27]	74.9	83.0	90.3	71.1	77.2	91.0	71.5	84.8	41
PointVector-XL [8]	78.4	86.1	91.9	72.3	78.1	91	-	58.5	24.1
PointNet++ [24]	54.5	67.1	81.0	53.5	-	83.0	-	-	-
+X-3D	62.9↑7.4	78.3↑11.2	84.4↑1.1	57.8↑4.3	-	85.3↑2.3	-	-	-
PointMetaBase-L [17]	75.6	-	90.6	69.7	-	90.7	71.0	2.0	2.7
+X-3D	76.7↑1.1	-	91.1↑0.5	71.9↑2.2	-	91.2↑0.6	71.8↑0.8	2.2	3.8
PointMetaBase-XL [17]	76.3	-	91.0	71.6	-	90.6	71.8	9.2	15.3
+X-3D	77.7↑1.4	-	91.6↑0.6	72.1↑0.5	-	91.4↑0.8	72.8↑1.0	9.8	18.1
DeLA [2]	78.3	86.1	91.7	74.1	80.0	92.2	75.9	14.0	7.0
+X-3D	**79.2**↑0.9	**86.5**↑0.4	**91.9**↑0.2	**74.3**↑0.2	**80.1**↑0.1	**92.2**	**76.3**↑0.4	14.8	8.0

the neighborhood context to supplement the complete local region information. Some previous works [12–14] such as RandLA [12] by multiple local feature aggregation, LCPFormer [13] by collecting features of neighborhood points in different neighborhoods to propagate local context. These practices often expand the receptive field and fuse local structures that are far away.

Our method generates the dynamic kernel from the explicit geometric structure of the current neighborhood. However, distant local structures might reside on different underlying manifolds, causing conflicts in geometric information. To address this, we limit context propagation to consider solely the local neighborhoods close in geometric properties to the center point, minimizing conflicts.

Due to the randomness of kNN and ball query, the constructed neighborhood usually produces overlapping parts. Therefore, for the center point $\mathbf{p}_i$ of a neighborhood, it may also be a neighborhood point $\mathbf{p}_{*,i}$ of other neighborhoods at the same time. We use this overlap phenomenon to fuse the features of the i-th point in different situations to propagate the context of different neighborhoods. If the i-th point is a center point, its feature $\hat{\mathbf{f}}_i$ is obtained as follows:

$$\hat{\mathbf{f}}_i = MaxPool(\hat{\mathbf{f}}_{i,j}), j = 1, 2...k \tag{16}$$

Then if the i-th point is a neighborhood point in the neighborhood centered at j-th point, its feature $\hat{\mathbf{f}}_{j,i}$ is obtained from (15). However, a point may be simultaneously captured as a neighborhood point by more than one neighborhood, so when the i-th point is a neighborhood point, we define the set of neighborhoods containing i-th point as $\mathcal{G}_i$. So for each element j in $\mathcal{G}_i$, there will be $i \in \mathcal{N}(\mathbf{p}_j)$, where $\mathcal{N}(\mathbf{p}_j)$ represents the neighborhood centered at j-th point. Thus we will aggregate the features at i-th point in $\mathcal{G}$ representing the local structural context:

Table 3. Classification results on the ScanObjectNN dataset. X-3D only adds an additional 0.1 M and 0.1 GFLOPs for a 0.7% OA improvement, achieving new state-of-the-art performance (**rank1 without extra training data**).

Method	**ScanObjectNN**		GFLOPs	#Params
	OA	mAcc		
DGCNN [34]	78.1	73.6	2.43	1.82M
PointNet [23]	68.2	63.4	0.45	3.47M
PointNet++ [24]	77.9	75.4	1.7	1.5M
MVTN [10]	82.8	-	1.78	4.24M
SPoTr [22]	88.6	86.8	-	-
PointMLP [20]	85.7	84.4	31.4	13.2M
PointNeXt-S [27]	88.20	86.84	1.64	1.4M
PointVector [8]	88.17	86.69	-	1.55M
PointMetaBase-S [17]	88.2	86.90	0.55	1.4M
+X-3D	88.83↑0.63	87.16↑0.26	0.59	1.5M
DeLA [2]	90.4	89.3	1.50	5.3M
+X-3D	**90.7**↑0.3	**89.9**↑0.6	1.53	5.4M

$$\hat{\mathbf{f}}_i^{(1)} = \frac{1}{Len(\mathcal{G}_i)} \sum_j \hat{\mathbf{f}}_{j,i}, j \in \mathcal{G}_i \tag{17}$$

Finally, we dynamically fuse the current neighborhood features with the context:

$$\hat{\mathbf{f}}_i^{(2)} = MLP([\hat{\mathbf{f}}_i^{(1)}, \hat{\mathbf{f}}_i]) \tag{18}$$

5. Experiments

We evaluated X-3D on four tasks: segmentation, classification, object part segmentation, and detection to prove its effectiveness. At the same time, we embedded X-3D on the simplest and most advanced models to demonstrate its applicability. Further, we designed a series of ablation experiments to demonstrate the effectiveness of various parts

Table 4. Detection results on the ScanNetV2 [5] and the SUN RGB-D [31] dataset. We tested the performance of X-3D on GroupFree [19] and report on mAP@0.25 and mAP@0.50.

Method	BackBone	ScanNetV2		SUN RGB-D		#Param	GFLOPs
		mAP@0.25	mAP@0.50	mAP@0.25	mAP@0.50		
VoteNet [25]	PointNet++	62.9	39.9	59.1	35.8	-	-
ImVoteNet [26]	PointNet++	-	-	63.4	-	-	
H3DNet [42]	4× PointNet++	67.2	48.1	60.1	39.0	-	-
3DETR [21]	Transformer	65.0	47.0	59.1	32.7	-	-
BRNet [3]	PointNet++	66.1	50.9	61.1	43.7	-	-
GroupFree^{6,256}	PointNet++	67.3	48.9	63.0	45.2	11.49	6.7
GroupFree^{6,256}	RepSurf-U	68.8	50.5	**64.3**	45.9	11.50	6.8
GroupFree^{6,256}	**X-3D**	**69.0**↑1.7	**51.1**↑2.2	**64.5**↑1.5	**46.9**↑1.7	12.30	7.1

Table 5. 3D Outdoor Object Detection Results on KITTI dataset on 3DSSD. We report the mIoU at three levels of difficulty

Method	easy	moderate	hard
3DSSD+**X-3D**	87.6(**88.5**)	78.1(**78.3**)	76.7(**76.9**)

of X-3D. In the supplementary material, there will be more detailed experiments .

5.1. Segmentation

We evaluated X-3D on two datasets for scene segmentation, S3DIS [1] and ScanNet [5].

S3DIS. There are six large indoor scenes, which contain 271 rooms and 13 categories in S3DIS. Based on the generic setting, we evaluated X-3D on 6-Fold and Area5 and reported the mIoU, mAcc and OA. As shown in Table 2, X-3D can effectively improve its performance on different scale models such as PointNet++ [24], PointMetaBase [17], DeLA [2], and the extra computational overhead is acceptable. For PointNet++, X-3D improves 4.3% and 7.4% mIoU on Area5 and 6-Fold, respectively. For PointMetaBase, We evaluated X-3D on its lightweight version PointMetaBase-L as well as its large-scale version PointMetaBase-XL. Although PointMetaBase-L is small in scale, it only brings 1.1 M and 0.2 GFLOPs extra overhead by embedding X-3D and improves 2.2% and 1.1% mIoU on Area 5 and 6-Fold, exceeding the XL version. And for PointMetaBase-XL, X-3D improves 0.5% and 1.4% mIoU on Area5 and 6-Fold. Finally, we tested X-3D on DeLA, which improves 0.2% and 0.9% mIoU on Area 5 and 6-Fold and achieves the new state-of-the-art performance.

ScanNet. There are 1201 indoor training point clouds and 312 validation point clouds in ScanNet. We evaluate X-3D on PointMetaBase-L, PointMetaBase-XL and DeLA, which boost 0.8%,1.0%,0.4% mIoU respectively.

Through further analysis, we found that X-3D performs well on more complex datasets, such as S3DIS 6-Fold and ScanNet and simpler models such as PointNet++ and PointMetaBase-L, because it explicitly introduces an effective geometric prior, which greatly reduces the difficulty of model learning.

Table 6. Part segmentation results on the ShapeNetPart [39] dataset. We tested X-3D on the state-of-the-art model PointMetaBase [17], which can effectively improve the performance.

Method	ins. mIoU	cls. mIoU	Params	GFLOPs
PointNet++ [24]	85.1	81.9	1.0	4.9
KPConv [32]	86.4	85.1	-	-
PointTransformer [43]	86.6	83.7	7.8	-
PointMLP [20]	86.1	85.1	-	-
StratifiedFormer [14]	86.6	85.1	-	-
PointNeXt-S [27]	86.7	85.6	1.0	4.5
PointMetaBase-S (C=64) [17]	86.9	85.1	3.8	3.85
PointMetaBase-S (C=32) [17]	86.7	84.4	1.0	1.39
+X-3D	**87.0**↑0.3	**85.1**↑0.7	2.0	2.0

5.2. Classification

We evaluated X-3D on PointMetaBase-S [17] and DeLA [2] on ScanObjectNN [33] and reported mAcc and OA, which is shown in Table 3. There are 15,000 actual scanned objects divided into 15 classes with 2,902 unique object instances in ScanObjectNN. More importantly, since ScanObjectNN is a real-world dataset, it introduces background, noise and other factors, which increases the difficulty of model learning. Due to the efficiency of X-3D, adding only an additional 0.04 GFLOPs and 0.1 M parameters leads to a 0.7% OA improvement on PointMetaBase and 0.3%OA improvement on DeLA with only 0.03 GLLOPs and 0.1M parameters, achieving new state-of-the-art performance.

5.3. Indoor Detection

We evaluated X-3D on GroupFree^{6,256} [19] on two datasets ScanNetV2 [5] and SUN RGB-D [31] in Table 4. ScanNetV2 contains 1513 indoor scenes and 18 object classes. SUN RGB-D contains 5K indoor RGB and

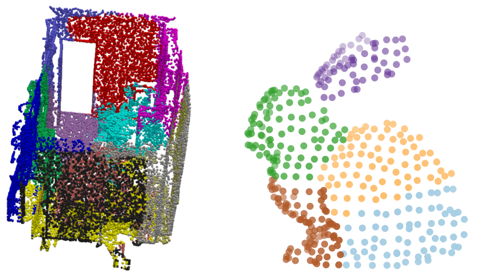


Figure 4. Kmeans Results of the Structure Kernel. The parameters of structure kerne are restricted by local structure, which provides a good structural prior.

depth images. Based on the generic setting, we report the mean Average Precision under the thresholds of 0.25 and 0.5. For ScanNetV2, X-3D improves 1.7%mAP@0.25 and 2.2% mAP@0.5. For SUN RGB-D, X-3D improves 1.5% mAP@0.25 and 1.7% mAP@0.50.

5.4. Outdoor Detection

As shown in Table 5, we applied X-3D to 3DSSD [38] and tested it on the KITTI [9] validation dataset. X-3D improves 0.9%,0.2%,0.2% mIoU on easy, moderate, and hard, respectively.

5.5. Part Segmentation

We evaluated X-3D on the state-of-the-art model PointMetaBase-S [17] on the ShapeNetPart [39] dataset, where the results are shown in Table 6. There are 16880 models with 16 different shape categories and 50 part labels in ShapeNetPart. Based on the generic setting, we report the final performance using the voting strategy. For PointMetaBase-S, X-3D improves 0.3% ins. mIoU and 0.7%cls. mIoU.

5.6. Analysis

In this section, we analyze in detail why X-3D works and compare it with Transformer.

Why X-3D works? We believe that the essence of X-3D effectiveness is similar to the principle of LLE methods in manifold learning. First, LLE limits the parameters of the mapping through the local structure (please refer to the supplementary material for details). Similarly, we argue that X-3D restricts the kernel parameter by its explicit structure, which greatly reduces the difficulty of model learning. To prove the validity of the restriction, we visualize the distribution of structure kernels and classify kernels with k-means in Figure 4. We find that the structure kernel is closely related to the geometric structure, such as the ears, head, feet and tail of the object in Figure 4b, as well as the walls, chairs and floors in the scene in Figure 4a. Therefore, the structure kernel contains explicit structural information to provide effective geometric priors for the model.

Table 7. We embed X-3D into Point Transformer by generating relational position embedding with our X-3D module and report the mIoU on S3DIS and ScanNet

	S3DIS	ScanNet
Point Transformer	70.4	70.6
+X-3D	71.6 (+1.2)	73.9 (+3.3)

Table 8. We compare computational cost between X-3D and Transformer.

	c=256,k=16	c=256,k=32	c=512,k=16
X-3D (structure kernel)	0.24	0.34	0.46
PTv1 (vector attention)	1.51	2.21	5.83
VIT (scalar attention)	0.84	1.02	3.29

Comparison with Transformer. Firstly, there are some differences between Point Transformer [14, 43] and implict high-dimensional structure modeling. Although the transformer dynamically generates the kernel corresponding to each relation vector, it considers local structural information during the generation process. For example, the process of PointTransformer [43] can be written as:

$$M_{i,j} = MLP(\mathbf{f}_i - \mathbf{f}_{i,j}) + MLP(\mathbf{p}_i - \mathbf{p}_{i,j}) \tag{19}$$

$$IS_i = \sum_{j=1}^{k} exp(M_{i,j}) \tag{20}$$

$$\mathbf{W}_{i,j} = \theta(M_{i,j}, IS_i) \tag{21}$$

Compare to the (21) with (7), the main difference between Transformer and X-3D is that we use the explicit structure ES_i directly captured in the data while Transformer uses the implicit structure IS_i learned by the network. We argue that the explicit local structure greatly reduces the difficulty of model learning. In Table 2, we have performed a performance comparison of X-3D with PointTransformer. In this case, we argue that X-3D is fundamentally different from Transformers and can be applied to Transformers as well. Therefore, we embedded X-3D into Point Transformer by generating relational position embedding with our X-3D module instead of the original linear projection and the results are reported in Table 7.

More importantly, compared with transformer-generating vector kernels for a single vector, X-3D generates a shared dynamic structure kernel for all vectors in a local neighborhood, which reduces a significant amount of computational cost. Here we provide the comparison between the computational cost in further detail. For an input of shape $N * K * (C + 3)$, where N is the number of points, C is the feature channel number, and K is the number of neighborhood points. We set different C and K

Table 9. We tested the effect of different utilization of ES_i. We report the Euclidean distance (GAP) between the local structure of the embedding space and the input space and the OA.

	GAP	OA	#Params	GFLOPs
baseline	5.3	88.2	1.4	0.55
Concat	3.1	88.3	1.4	0.58
Vector Kernel	1.6	88.6	1.5	0.83
Structure Kernel	**1.3**	**88.8**	1.5	0.59

and report the results (GFLOPs). From Table 8, we see that X-3D can save a lot of computation costs.

5.7. Ablation Study

Structure Kernel. As shown in Table 9, we tested the effect of different utilization of ES_i on ScanObjectNN. In the first row we report the performance of the original model. Then in the second row, we try to directly merge the explicit local structure with the relation vector. Although the explicit introduction of the local structure information reduces the GAP, the simple merging cannot model the relation vector through the strong restriction of the local structure, resulting in limited performance improvement. Further, in the third row, we try to generate vector kernels according to the correlation between each relation vector and the explicit local structure which is similar to transformer. Although this method strongly restricts the modeling of relation vectors by the explicit local structure and has higher performance, it also brings a large computational cost. Finally, in the last row, we report the performance of structure kernel, since it directly generates a shared kernel for all relation vectors according to the local structure, the computational cost and performance is greatly reduced and improved.

Denoising and Neighborhood Context Propagation. As shown in Table 10, we test the effect of denoising step and neighborhood context propagation step on PointMetaBase-L on S3DIS. For denoising step, noise points may distort local structural information, which is also discussed in RepSurf and other methods, so the denoising process is important, and we see that it can improve performance on various datasets and tasks. For neighborhood context propagation step, its purpose is to solve the problem caused by randomly selecting neighborhood points. However, for the classification task, the number of points is small, so the influence of neighborhood randomness is small. It can be seen that this step has little improvement for the classification task, but has a great performance improvement for the segmentation task with a large number of points.

Robustness. Since X-3D inherits the advantages of LLE and is more robust to local transformation, we report the performance of different models on ScanObjectNN before and after transformation in Table 11. We see that X-3D is more robust because the introduction of the explicit structure strengthens the feature extraction ability of the model for different structures.

Table 10. We test the effect of the denoising step (**DN**) and the neighborhood context propagation (**NCP**) step on PointMetaBase on S3DIS Area5 and ScanObjectNN.

		+ DN	+ NCP	+ DN&NCP
S3DIS (mIoU%)	70.9	71.3	71.4	71.9
ScanObjectNN (OA%)	88.5	88.8	88.6	88.8

Table 11. We test the robustness of X-3D on ScanObjectNN with different transformations.

	Vanilla	Rotation			Scale	
		30	60	90	0.9	1.1
X-3D	88.8	88.6 (-0.2)	88.3 (-0.5)	88.5 (-0.3)	88.6 (-0.2)	88.7 (-0.1)
PointMetaBase	88.2	87.4 (-0.8)	87.1 (-1.1)	87.7 (-0.5)	87.4 (-0.8)	87.5 (-0.7)

5.8. Visualization

As shown in Fig 5, we visualize the segmentation results on S3DIS and can see that X-3D segmentation prediction is generally more complete and accurate due to the introduction of explicit local structure information.

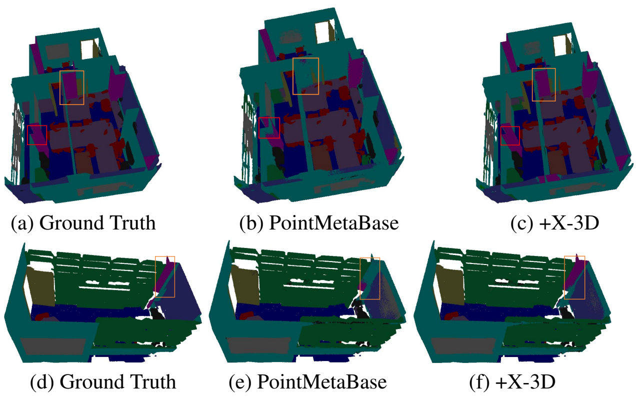


Figure 5. We visualize the segmentation results of PointMetaBase-L as well as X-3D on S3DIS Area5 and annotate the different places

6. Conclusion

In this paper, we present X-3D, an explicit 3D structure-based modeling approach which uses an explicit input structure to limit the embedding parameters. Experimental results show that X3D can effectively reduce the difficulty of model learning and achieve the latest state-of-the-art performance

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China under Grant 62376032 and Grant U22B2050.

References

[1] I Armeni, S Sax, AR Zamir, S Savarese, A Sax, AR Zamir, and S Savarese. Joint 2d-3d-semantic data for indoor scene understanding. arxiv 2017. *arXiv preprint arXiv:1702.01105*, 2017. 5, 6

[2] Binjie Chen, Yunzhou Xia, Yu Zang, Cheng Wang, and Jonathan Li. Decoupled local aggregation for point cloud learning. *arXiv preprint arXiv:2308.16532*, 2023. 5, 6

[3] Bowen Cheng, Lu Sheng, Shaoshuai Shi, Ming Yang, and Dong Xu. Back-tracing representative points for voting-based 3d object detection in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8963–8972, 2021. 6

[4] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 2

[5] 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. 5, 6

[6] Jérôme Demantké, Clément Mallet, Nicolas David, and Bruno Vallet. Dimensionality based scale selection in 3d lidar point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38:97–102, 2012. 4

[7] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1201–1209, 2021. 2

[8] Xin Deng, WenYu Zhang, Qing Ding, and XinMing Zhang. Pointvector: A vector representation in point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9455–9465, 2023. 5

[9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 7

[10] 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. 5

[11] Haotian Hu, Fanyi Wang, Zhiwang Zhang, Yaonong Wang, Laifeng Hu, and Yanhao Zhang. Gam: gradient attention module of optimization for point clouds analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 835–843, 2023. 1, 2

[12] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020. 1, 2, 5

[13] Zhuoxu Huang, Zhiyou Zhao, Banghuai Li, and Jungong Han. Lcpformer: Towards effective 3d point cloud analysis via local context propagation in transformers. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 5

[14] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8500–8509, 2022. 2, 5, 6, 7

[15] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4558–4567, 2018. 4

[16] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9267–9276, 2019. 2

[17] Haojia Lin, Xiawu Zheng, Lijiang Li, Fei Chao, Shanshan Wang, Yan Wang, Yonghong Tian, and Rongrong Ji. Meta architecture for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17682–17691, 2023. 1, 2, 5, 6, 7

[18] 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. 1, 2, 3

[19] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2949–2958, 2021. 6

[20] 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. *International Conference on Learning Representations*, 2022. 5, 6

[21] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021. 6

[22] Jinyoung Park, Sanghyeok Lee, Sihyeon Kim, Yunyang Xiong, and Hyunwoo J Kim. Self-positioning point-based transformer for point cloud understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21814–21823, 2023. 5

[23] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 5

[24] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30, 2017. 1, 2, 5, 6

[25] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 6

[26] Charles R Qi, Xinlei Chen, Or Litany, and Leonidas J Guibas. Imvotenet: Boosting 3d object detection in point clouds with image votes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4404–4413, 2020. 6

[27] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 35:23192–23204, 2022. 1, 5, 6

[28] Haoxi Ran, Jun Liu, and Chengjie Wang. Surface representation for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18942–18952, 2022. 2, 5

[29] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290 (5500):2323–2326, 2000. 2, 4

[30] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3693–3702, 2017. 2

[31] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 567–576, 2015. 6

[32] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6411–6420, 2019. 1, 2, 3, 5, 6

[33] 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. 6

[34] 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. 2, 5

[35] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35:33330–33342, 2022. 5

[36] 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. 2

[37] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 2

[38] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11040–11048, 2020. 7

[39] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 6, 7

[40] Min Zhang, Haoxuan You, Pranav Kadam, Shan Liu, and C-C Jay Kuo. Pointhop: An explainable machine learning method for point cloud classification. *IEEE Transactions on Multimedia*, 22(7):1744–1755, 2020. 4

[41] Zhenyue Zhang and Hongyuan Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal on Scientific Computing*, 26(1):313–338, 2004. 2

[42] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primitives. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 311–329. Springer, 2020. 6

[43] 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. 1, 2, 5, 6, 7

[44] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 2