Clustering based Point Cloud Representation Learning for 3D Analysis
Supplemental Material

Tuo Feng[1], Wenguan Wang[2], Xiaohan Wang[2], Yi Yang[2]*, Qinghua Zheng[3]

[1] ReLER, AAII, University of Technology Sydney [2] ReLER, CCAI, Zhejiang University [3] Xi'an Jiaotong University

https://github.com/FengZicai/Cluster3Dseg/

In this supplementary material, we provide the following sections for a better understanding of the main paper. The pseudo-code of clustering based point cloud segmentation learning is elaborated in §A. §B presents the distribution of point data over cluster centers. More qualitative and quantitative results are further presented and analyzed in §C. Finally, limitation and societal impact are discussed in §D.

A. Pseudo-Code

Algorithm S1 provides a pseudo-code of 'assigning subclass labels' function and 'update operation' function. Correspondingly, Algorithm S2 provides a pseudo-code of $\mathcal{J}_{\text{PCC}}$ (see Eq. (7)). The implementation of $\mathcal{J}_{\text{PPC}}$ is similar to it, so we do not show pseudo-code for $\mathcal{J}_{\text{PPC}}$. Moreover, to guarantee the reproducibility, our code is released at: `github.com/FengZicai/Cluster3Dseg`.

B. Distribution of Subclass Clusters

Fig. S1 shows point assignment distribution for 'truck' and 'traffic-sign' classes, with different numbers $M = \{10, 20, 40, 60, 80\}$ of clusters. We can find that 1) the number of point samples assigned to each cluster is different; 2) with the increase of M, some sub-class centers only contain a limited number of samples, especially when $M = 80$. In this case, the value of M has exceeded the number of underlying subclass centers in the dataset, resulting in over-clustering. And therefore, some trivial patterns may distract the model and cause performance degradation.

C. More Qualitative and Quantitative Results

Complete Quantitative Result on SemanticKITTI Single-Scan Challenge `test`. Table S1 and S2 report the complete results on SemanticKITTI [1] single-scan challenge `test`. Our method reaches 70.4% mIoU, which yields 2.6% mIoU gains over Cylinder3D [2]. Moreover, it also outperforms many famous segmentation models, such as AF2S3Net [3] and RPVNet [4]. One more thing to point out, spvnas[1] did not provide the source code of 3D-NAS pipeline and the control file for $\text{SPVNAS}_{12.5\text{M}}$. But the control file and pre-trained models for $\text{SPVNAS}_{10.8\text{M}}$ are shared[2]. And the difference between $\text{SPVNAS}_{12.5\text{M}}$ and $\text{SPVNAS}_{10.8\text{M}}$ is that $\text{SPVNAS}_{10.8\text{M}}$ is trained except sequence 08. As for our implementation, $\text{SPVNAS}_{10.8\text{M}}$ and $\text{SPVNAS}_{10.8\text{M}}$ + **`Ours`** are trained on sequences 00-10 and evaluated on 11-21.

Complete Quantitative Result on S3DIS Area-5. Table S3 and S4 present the complete per-class IoU on S3DIS [5] Area-5. Both CBL[6] and our method use contrastive loss on the premise of fully supervised learning. But [6] only samples negative points locally around the boundaries, while we contrast global subclass centers against the points sampled from the ENTIRE training dataset. Our idea is much more powerful and insightful. The fair comparison based on PTV1[7] shows that our approach attains mIoU/mAcc of 72.2%/79.6%, outperforming PTV1+CBL (71.6%/77.9%).

Complete Quantitative Result on SemanticKITTI multi-scan challenge `test`. Table S5 and S6 report the complete results on SemanticKITTI [1] multi-scan challenge `test`. With Cylinder3D, our algorithm also attains consistent performance improvements of 2.2% mIoU, just like that in single-scan `test`. Moreover, Cylinder3D+ **`Ours`** surpasses Cylinder3D in 17 classes out of 25 classes.

Qualitative Results for Segmetation. We show more qualitative results on SemanticKITTI [1] single-scan challenge `val` (Fig. S2), S3DIS [5] Area-5 (Fig. S3) and SemanticKITTI [1] multi-scan challenge `val` (Fig. S4). As observed, our approach generally gives more accurate predictions compared with vanilla PTV1[7] and Cylinder3D[2]. In Fig. S3, vanilla PTV1 fails to recognize region boundaries and tends to misclassify board-like objects, while our method can significantly reduce these errors. Fig. S4 depicts qualitative comparisons of Cylinder3D and Cylinder3D + **`Ours`** over lidar sequences on SemanticKITTI multi-scan challenge `val`. Note that, the predicted labels of five consecutive frames are displayed in one frame. It can be observed

[1] https://github.com/mit-han-lab/spvnas/

[2] SPVNAS has cancelled the download link for the Control file and $\text{SPVNAS}_{10.8\text{M}}$ model. Instead, we will release the two previously downloaded files.

that Cylinder3D + **Ours** has smaller errors over the semantic boundaries as well as classes belonging to ground and nature.

D. Limitation and Societal Impact

License of Assets. Cylinder3D[3] is released with Apache license. KPConv[4] is implemented based on its released code with MIT license. We have also implemented our method on Point Transformer[5]. SPVNAS[6] is implemented based on its released code with MIT license. For 3D object detection, our implementation is based on OpenPCDet[7], and it is released under the Apache 2.0 license.

Limitation. For some very rare classes, such as *beam* in S3DIS [5], *bicyclist* in SemanticKITTI [1] multi-scan challenge, our algorithm did not show better-improved results. However, many previous state-of-the-arts [2, 7, 8] also perform poorly on these classes. In the future, we plan to explore smarter data sampling strategies and hard example synthesis techniques to address this issue.

Societal Impact. For the potential negative societal impacts, in real-world robot navigation tasks or autonomous driving tasks, inaccurate prediction of point cloud labels may lead agents to the wrong category and raise human safety concerns. To avoid this potential problem, we suggest proposing a security protocol in case of dysfunction of our algorithm in real-world applications.

[3] https://github.com/xinge008/Cylinder3D/

[4] https://github.com/HuguesTHOMAS/KPConv-PyTorch/

[5] https://github.com/POSTECH-CVLab/point-transformer

[6] https://github.com/mit-han-lab/spvnas/

[7] https://github.com/HuguesTHOMAS/KPConv-PyTorch/

Algorithm S1 Pseudo-code of clustering based point cloud segmentation learning - Part I.

```
# M: number of subclusters.
# nc: number of classes.
# dim: number of dimensions.
# x: features (N, dim).
# ŷ: labels (N).
# y: predicted labels (N).
# cc: cluster centers (num_classes, M, dim).
# L: clustering results.
# μ: momentum coefficient.

def _assigning_subclass_labels(x, ŷ, y):
    # selected features, subclass labels.
    # selected cluster center embbedings, cluster center labels.
    X_o,y_o,X̃_o,ỹ_o = [],[],[],[]
    # Record of new cluster centers for this iteration, see Eq.(5).
    ncc = zeros(nc, M, dim)
    this_class = unique(this_y)
    for idx in this_class:
        indices = (ŷ == idx).nonzero()
        # select cluster centers with idx.
        pc = select(cc, idx)
        # select features with indices.
        xc = select(x, indices)
        yc = select(ŷ, indices)
        PS = mm(xc, pc.T)
        PS = softmax(PS, 1)
        # Sinkhorn-Knopp algorithm.
        online_clustering()
        yc = yc * M
        yc = yc + L
        # Averageing xc tensor according to L.
        ncc[idx] = scatter_mean(xc, L, dim=0, dim_size=M)
        # append to output variables.
        X_o = append(X_o, xc)
        y_o = append(y_o, yc)
        X̃_o = append(X̃_o, pc)
        ỹ_o = append(ỹ_o, idx.repeat(M) * M+tensor(list(range(M))))
    return X_o,y_o,X̃_o,ỹ_o

def _update_operation():
    cc = cc * μ + ncc * (1 - μ)
    cc = normalize((cc, p=2, dim=2))
    return cc
```

Algorithm S2 Pseudo-code of clustering based point cloud segmentation learning - Part II.

```
# temperature: scalar temperature parameter
# Xᵢ: selected features
# yᵢ: selected subclass labels
# X̃ᵢ: cluster center embbedings
# ỹᵢ: cluster center labels

def _pcc_contrastive(Xᵢ,yᵢ,X̃ᵢ,ỹᵢ):
    anchor_label = yᵢ.view(-1, 1)
    contrast_label = ỹᵢ.view(-1, 1)
    anchor_feature = Xᵢ
    contrast_feature = X̃ᵢ

    mask = eq(anchor_label, contrast_label.T)
    anchor_dot_contrast = div( matmul(anchor_feature,
                              contrast_feature.T), temperature)
    logits_max, _ = max(anchor_dot_contrast, dim=1, keepdim=True)
    # To avoid the numerical overflow
    logits = anchor_dot_contrast - logits_max.detach()

    # neg_logits mean the sum of logits of all negative pairs
    neg_mask = 1 - mask
    neg_logits = exp(logits) * neg_mask
    neg_logits = neg_logits.sum(1, keepdim=True)

    # exp_logits mean the logit of each sample pair
    exp_logits = exp(logits)
    log_prob = logits - log(exp_logits + neg_logits)
    mean_log_prob_pos = (mask * log_prob).sum(1) / mask.sum(1)

    loss = - (temperature / base_temperature) * mean_log_prob_pos
    loss = loss.mean()
    return loss
```

Figure S1: Distribution plot with different numbers $M = \{10, 20, 40, 60, 80\}$ of clusters for 'truck' and 'traffic-sign' classes. (Best viewed with zoom-in.)

Table S1: **Quantitative results** on SemanticKITTI [1] single-scan challenge `test` (§4.1) - Part I. mIoU (%) and IoUs (%) are reported.

Method	mIoU	road	sidewalk	parking	other-ground	building	car	truck	bicycle	motorcycle	other-vehicle
TangentConv$_{\text{CVPR18}}$[9]	40.9	83.9	63.9	33.4	15.4	83.4	90.8	15.2	2.7	16.5	12.1
SqueezeSegV2$_{\text{ICRA19}}$[10]	39.7	88.6	67.6	45.8	17.7	73.7	81.8	13.4	18.5	17.9	14.0
DarkNet53$_{\text{ICCV19}}$[1]	49.9	91.8	74.6	64.8	27.9	84.1	86.4	25.5	24.5	32.7	22.6
Rangenet++$_{\text{IROS19}}$[11]	52.2	91.8	75.2	65.0	27.8	87.4	91.4	25.7	25.7	34.4	23.0
3D-MiniNet$_{\text{IROS20}}$[12]	55.8	91.6	74.5	64.2	25.4	89.4	90.5	28.5	42.3	42.1	29.4
PointASNL$_{\text{CVPR20}}$[13]	46.8	87.4	74.3	24.3	1.8	83.1	87.9	39.0	0.0	25.1	29.2
PolarNet$_{\text{CVPR20}}$[14]	54.3	90.8	74.4	61.7	21.7	90.0	93.8	22.9	40.3	30.1	28.5
RandLA-Net$_{\text{CVPR20}}$[15]	55.9	90.5	74.0	61.8	24.5	89.7	94.2	43.9	29.8	32.2	39.1
SqueezeSegV3$_{\text{ECCV20}}$[16]	55.9	91.7	74.8	63.4	26.4	89.0	92.5	29.6	38.7	36.5	33.0
SalsaNext$_{\text{ISVC20}}$[17]	59.5	91.7	75.8	63.7	29.1	90.2	91.9	38.9	48.3	38.6	31.9
FusionNet$_{\text{ECCV20}}$[18]	61.3	91.8	77.1	68.8	30.8	92.5	95.3	41.8	47.5	37.7	34.5
JS3C-Net$_{\text{AAAI21}}$[19]	66.0	88.9	72.1	61.9	31.9	92.5	95.8	**54.3**	59.3	52.9	46.0
AF2S3Net$_{\text{CVPR21}}$[3]	69.7	91.3	72.5	68.8	**53.5**	87.9	94.5	39.2	65.4	**86.8**	41.1
RPVNet$_{\text{ICCV21}}$[4]	70.3	**93.4**	**80.7**	70.3	33.3	**93.5**	**97.6**	44.2	**68.4**	68.7	**61.1**
PVKD$_{\text{CVPR22}}$[20]	**71.4**	91.8	77.5	**70.9**	41.0	92.4	97.0	53.5	67.9	69.3	60.2
KPConv$_{\text{ICCV19}}$[8]	58.8	88.8	72.7	61.3	31.6	90.5	96.0	33.4	30.2	42.5	44.3
KPConv + `Ours`	61.0	89.9	75.0	63.4	34.3	91.4	88.8	49.0	45.0	46.6	45.5
SPVNAS$_{10.8\text{M}}$$_{\text{ECCV20}}$[21]	62.3	89.6	73.8	63.2	29.1	90.9	96.7	50.9	40.6	42.1	51.3
SPVNAS$_{10.8\text{M}}$ + `Ours`	64.3	89.6	73.9	64.0	28.8	91.4	96.7	48.0	48.9	50.5	51.0
Cylinder3D$_{\text{CVPR21}}$[2]	67.8	91.4	75.5	65.1	32.3	91.0	97.1	50.8	67.6	64.0	58.6
Cylinder3D + `Ours`	70.4	91.7	77.2	66.1	34.1	92.3	97.0	51.9	**68.4**	65.8	58.8

Table S2: **Quantitative results** on SemanticKITTI [1] single-scan challenge `test` (§4.1) - Part II. mIoU (%) and IoUs (%) are reported.

Method	mIoU	vegetation	trunk	terrain	person	bicyclist	motorcyclist	fence	pole	traffic-sign
TangentConv$_{\text{CVPR18}}$[9]	40.9	79.5	49.3	58.1	23.0	28.4	8.1	49.0	35.8	28.5
SqueezeSegV2$_{\text{ICRA19}}$[10]	39.7	71.8	35.8	60.2	20.1	25.1	3.9	41.1	20.2	26.3
DarkNet53$_{\text{ICCV19}}$[1]	49.9	78.3	50.1	64.0	36.2	33.6	4.7	55.0	38.9	52.2
Rangenet++$_{\text{IROS19}}$[11]	52.2	80.5	55.1	64.6	38.3	38.8	4.8	58.6	47.9	55.9
3D-MiniNet$_{\text{IROS20}}$[12]	55.8	82.8	60.8	66.7	47.8	44.1	14.5	60.8	48.0	56.6
PointASNL$_{\text{CVPR20}}$[13]	46.8	84.1	52.2	70.6	34.2	57.6	0.0	43.9	57.8	36.9
PolarNet$_{\text{CVPR20}}$[14]	54.3	84.0	65.5	67.8	43.2	40.2	5.6	61.3	51.8	57.5
RandLA-Net$_{\text{CVPR20}}$[15]	55.9	83.8	63.6	68.6	48.4	47.4	9.4	60.4	51.0	50.7
SqueezeSegV3$_{\text{ECCV20}}$[16]	55.9	82.0	58.7	65.4	45.6	46.2	20.1	59.4	49.6	58.9
SalsaNext$_{\text{ISVC20}}$[17]	59.5	81.8	63.6	66.5	60.2	59.0	19.4	64.2	54.3	62.1
FusionNet$_{\text{ECCV20}}$[18]	61.3	84.5	69.8	68.5	59.5	56.8	11.9	69.4	60.4	66.5
JS3C-Net$_{\text{AAAI21}}$[19]	66.0	84.5	69.8	67.9	69.5	65.4	39.9	70.8	60.7	68.7
AF2S3Net$_{\text{CVPR21}}$[3]	69.7	70.2	68.5	53.7	**80.7**	**80.4**	**74.3**	63.2	61.5	71.0
RPVNet$_{\text{ICCV21}}$[4]	70.3	86.5	**75.1**	71.7	75.9	74.4	43.4	**72.1**	64.8	61.4
PVKD$_{\text{CVPR22}}$[20]	**71.4**	86.5	73.8	**71.9**	75.1	73.5	50.5	69.4	64.9	61.4
KPConv$_{\text{ICCV19}}$[8]	58.8	84.8	69.2	69.1	61.5	61.6	11.8	64.2	56.4	47.4
KPConv + `Ours`	61.0	72.0	56.5	68.8	59.4	60.1	36.4	66.1	49.5	60.4
SPVNAS$_{10.8\text{M}}$$_{\text{ECCV20}}$[21]	62.3	85.5	70.3	69.8	60.4	62.8	21.8	65.3	57.6	62.0
SPVNAS$_{10.8\text{M}}$ + `Ours`	64.3	85.3	72.1	69.1	67.1	70.5	23.2	67.0	60.7	64.5
Cylinder3D$_{\text{CVPR21}}$[2]	67.8	85.4	71.8	68.5	73.9	67.9	36.0	66.5	62.6	65.6
Cylinder3D + `Ours`	70.4	**86.7**	73.5	71.7	69.6	70.1	54.6	70.8	**65.1**	**71.6**

Table S3: **Quantitative results** on S3DIS [5] Area-5 (§4.2) - Part I. mIoU (%) and IoUs (%) are reported.

Method	mIoU	mAcc	OA	ceiling	floor	wall	beam	column	window	door
PointNet[CVPR17][22]	41.1	49.0	-	88.8	97.3	69.8	0.1	3.9	46.3	10.8
SegCloud[3DV17][23]	48.9	57.4	-	90.1	96.1	69.9	0.0	18.4	38.4	23.1
TangentConv[CVPR18][9]	52.6	62.2	-	90.5	97.7	74.0	0.0	20.7	39.0	31.3
PointCNN[NeurIPS18][24]	57.3	63.9	85.9	92.3	98.2	79.4	0.0	17.6	22.8	62.1
SPGraph[CVPR18][25]	58.0	66.5	86.4	89.4	96.9	78.1	0.0	42.8	48.9	61.6
PCCN[CVPR18][26]	58.3	-	67.0	92.3	96.2	75.9	0.3	6.0	69.5	63.5
HPEIN[ICCV19][27]	61.9	68.3	87.2	91.5	98.2	81.4	0.0	23.3	65.3	40.0
PAT[CVPR19][28]	60.1	70.8	-	93.0	98.5	72.3	1.0	41.5	85.1	38.2
PointWeb[CVPR19][29]	60.3	66.6	87.0	92.0	98.5	79.4	0.0	21.1	59.7	34.8
MinkowskiNet[CVPR19][30]	65.4	71.7	-	91.8	98.7	86.2	0.0	34.1	48.9	62.4
SCF-Net[CVPR21][31]	63.8	-	-	-	-	-	-	-	-	-
BAAF-Net[CVPR21][32]	65.4	73.1	88.9	-	-	-	-	-	-	-
CGA-Net[CVPR21][33]	68.6	-	-	94.5	98.3	83.0	0.0	25.3	59.6	71.0
Stratified Trans.[CVPR22][34]	72.0	78.1	91.5	-	-	-	-	-	-	-
PTV2[NeurIPS22][35]	**72.6**	78.0	**91.6**	-	-	-	-	-	-	-
KPConv[ICCV19][8]	67.1	72.8	-	92.8	97.3	82.4	0.0	23.9	58.0	69.0
KPConv+ **Ours**	69.0	76.2	90.5	95.7	98.3	84.0	0.0	30.7	66.7	77.6
PTV1[ICCV21][7]	70.4	76.5	90.8	94.0	98.5	86.3	0.0	38.0	63.4	74.3
PTV1+CBL[CVPR22][6]	71.6	77.9	91.2	-	-	-	-	-	-	-
PTV1+ **Ours**	72.2	**79.6**	91.2	94.2	98.4	88.1	0.0	49.3	65.3	79.4

Table S4: **Quantitative results** on S3DIS [5] Area-5 (§4.2) - Part II. mIoU (%) and IoUs (%) are reported.

Method	mIoU	mAcc	OA	table	chair	sofa	bookcase	board	clutter
PointNet[CVPR17][22]	41.1	49.0	-	52.6	58.9	40.3	5.9	26.4	33.3
SegCloud[3DV17][23]	48.9	57.4	-	70.4	75.9	40.9	58.4	13.0	41.6
TangentConv[CVPR18][9]	52.6	62.2	-	77.5	69.4	57.3	38.5	48.8	39.8
PointCNN[NeurIPS18][24]	57.3	63.9	85.9	74.4	80.6	31.7	66.7	62.1	56.7
SPGraph[CVPR18][25]	58.0	66.5	86.4	84.7	75.4	69.8	52.6	2.1	52.2
PCCN[CVPR18][26]	58.3	-	67.0	66.9	65.6	47.3	68.9	59.1	46.2
HPEIN[ICCV19][27]	61.9	68.3	87.2	75.5	87.7	58.5	67.8	65.6	49.4
PAT[CVPR19][28]	60.1	70.8	-	57.7	83.6	48.1	67.0	61.3	33.6
PointWeb[CVPR19][29]	60.3	66.6	87.0	76.3	88.3	46.9	69.3	64.9	52.5
MinkowskiNet[CVPR19][30]	65.4	71.7	-	81.6	89.8	47.2	74.9	74.4	58.6
SCF-Net[CVPR21][31]	63.8	-	-	-	-	-	-	-	-
BAAF-Net[CVPR21][32]	65.4	73.1	88.9	-	-	-	-	-	-
CGA-Net[CVPR21][33]	68.6	-	-	82.6	92.2	77.7	76.4	69.5	61.5
Stratified Trans.[CVPR22][34]	72.0	78.1	91.5	-	-	-	-	-	-
PTV2[NeurIPS22][35]	**72.6**	78.0	**91.6**	-	-	-	-	-	-
KPConv[ICCV19][8]	67.1	72.8	-	81.5	91.0	75.4	75.3	66.7	58.9
KPConv+ **Ours**	69.0	76.2	90.8	79.9	91.0	70.3	76.7	63.0	63.6
PTV1[ICCV21][7]	70.4	76.5	90.8	89.1	82.4	74.3	80.2	76.0	59.3
PTV1+CBL[CVPR22][6]	71.6	77.9	91.2	-	-	-	-	-	-
PTV1+ **Ours**	72.2	**79.6**	91.2	89.4	82.2	74.8	77.6	81.0	58.7

Table S5: **Quantitative results** on SemanticKITTI [1] multi-scan challenge `test` (§4.3) - Part I. mIoU (%) and IoUs (%) are reported.

Method	mIoU	road	sidewalk	parking	other-ground	building	car	moving car	truck	moving truck	bicycle	motorcycle	other-vehicle	moving other-vehicle
TangentConv[CVPR18][9]	34.1	83.9	64.0	38.3	15.3	85.8	84.9	40.3	21.1	1.1	2.0	18.2	18.5	6.4
DarkNet53[ICCV19][1]	41.6	91.6	75.3	64.9	27.5	85.2	84.1	61.5	20.0	14.1	30.4	32.9	20.7	15.2
TemporalLidarSeg[3DV20][36]	47.0	**91.8**	75.8	59.6	23.2	89.8	92.1	68.2	39.2	2.1	47.7	40.9	35.0	12.4
SpSeqnet[CVPR20][37]	43.1	90.1	73.9	57.6	27.1	91.2	88.5	53.2	29.2	**41.2**	24.0	26.2	22.7	**26.2**
KPConv[ICCV19][8]	51.2	86.5	70.5	58.4	26.7	90.8	93.7	69.4	42.5	5.8	44.9	47.2	38.6	4.7
KPConv+ **Ours**	53.2	90.4	75.2	62.1	25.1	91.8	**95.8**	75.2	**43.8**	4.1	67.2	63.1	**44.2**	0.7
Cylinder3D[CVPR21][2]	52.5	90.7	74.5	65.0	**32.3**	**92.6**	94.6	74.9	41.3	0.0	**67.6**	**63.8**	38.8	0.1
Cylinder3D+ **Ours**	**54.7**	91.4	**76.9**	**66.1**	27.8	91.4	95.3	**81.7**	42.7	11.9	55.9	52.9	38.7	11.2

Table S6: **Quantitative results** on SemanticKITTI [1] multi-scan challenge `test` (§4.3) - Part II. mIoU (%) and IoUs (%) are reported.

Method	mIoU	vegetation	trunk	terrain	person	moving person	bicyclist	moving bicyclist	motorcyclist	moving motorcyclist	fence	pole	traffic-sign
TangentConv[CVPR18][9]	34.1	79.5	43.2	56.7	1.6	1.9	0.0	30.1	0.0	42.2	49.1	36.4	31.2
DarkNet53[ICCV19][1]	41.6	78.4	50.7	64.8	7.5	0.2	0.0	28.9	0.0	37.8	56.5	38.1	53.3
TemporalLidarSeg[3DV20][36]	47.0	82.3	62.5	64.7	14.4	40.4	0.0	42.8	0.0	12.9	63.8	52.6	60.4
SpSeqnet[CVPR20][37]	43.1	84.0	66.0	65.7	6.3	36.2	0.0	2.3	0.0	0.1	66.8	50.8	48.7
KPConv[ICCV19][8]	51.2	84.6	70.3	66.0	**21.6**	**67.5**	0.0	67.4	0.0	**47.2**	64.5	57.0	53.9
KPConv+ **Ours**	53.2	85.4	71.1	69.3	10.7	72.1	0.0	**68.5**	**9.9**	9.9	**67.5**	62.6	64.6
Cylinder3D[CVPR21][2]	52.5	85.8	72.0	68.9	12.5	65.7	**1.7**	68.3	0.2	11.9	66.0	63.1	61.4
Cylinder3D+ **Ours**	**54.7**	**86.5**	**72.7**	**71.6**	15.5	61.8	0.0	68.2	3.0	46.0	66.1	**64.0**	**68.0**

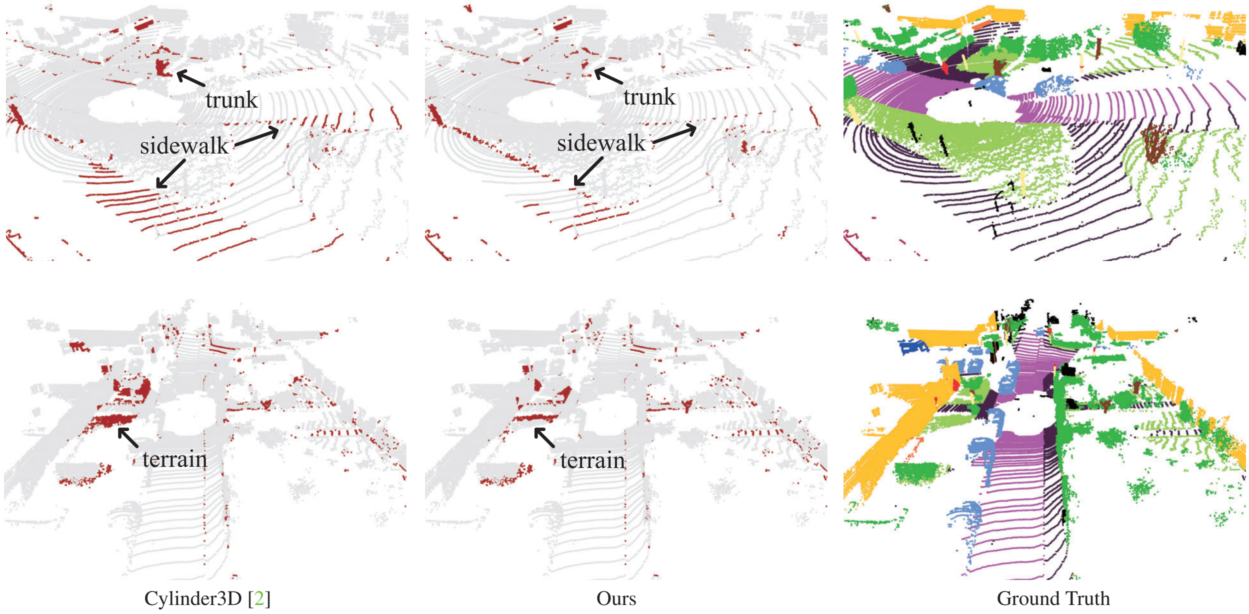

Figure S2: Error maps of Cylinder3D [2] and Ours on SemanticKITTI [1] single-scan challenge `val` (§4.1). The differences are as illustrated by arrows.

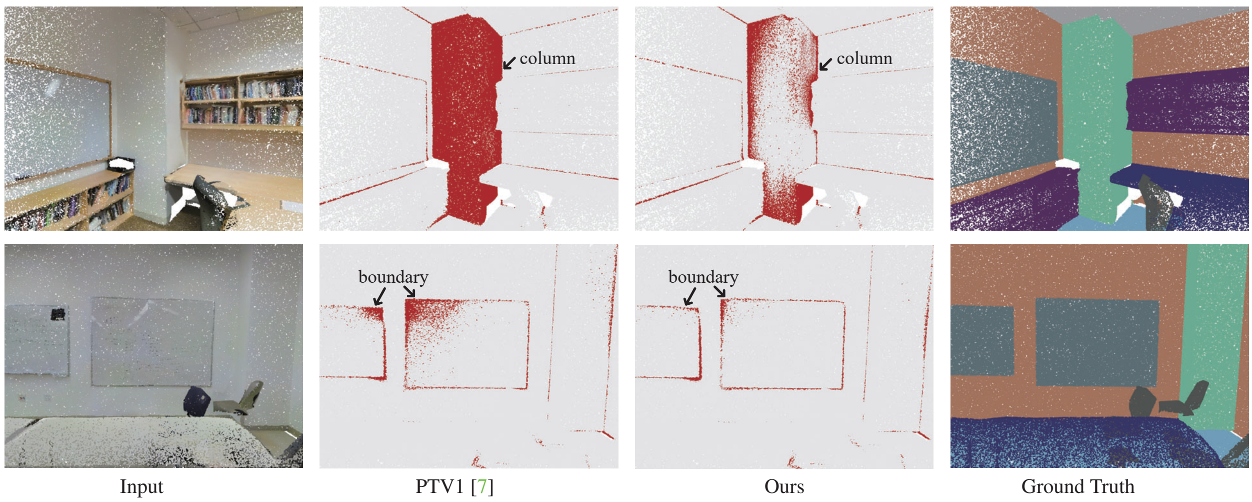

Figure S3: Error maps of PTV1 [7] and Ours on S3DIS [5] Area-5 (§4.2). The differences are as illustrated by arrows.

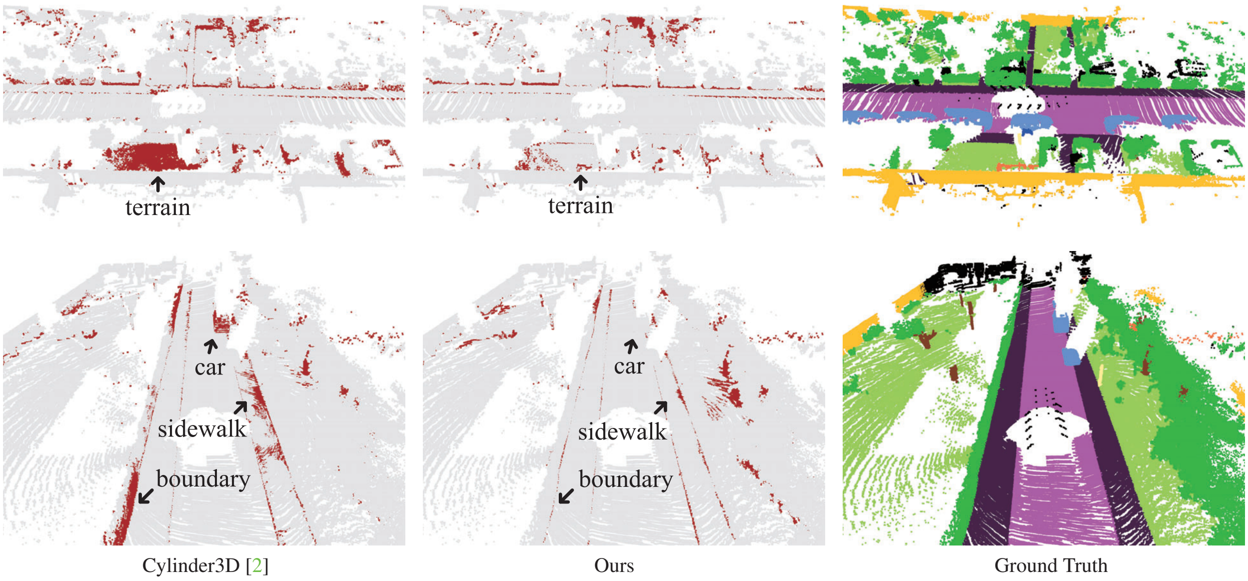

Figure S4: Error maps of Cylinder3D [2] and Ours on SemanticKITTI [1] multi-scan challenge `val` (§4.3). The differences are as illustrated by arrows.

References

[1] 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 *ICCV*, 2019. S1, S2, S6, S8, S9, S10

[2] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *CVPR*, 2021. S1, S2, S6, S8, S9, S10

[3] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. (af)2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In *CVPR*, 2021. S1, S6

[4] Jianyun Xu, Ruixiang Zhang, Jian Dou, Yushi Zhu, Jie Sun, and Shiliang Pu. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *ICCV*, 2021. S1, S6

[5] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *CVPR*, 2016. S1, S2, S7, S9

[6] Liyao Tang, Yibing Zhan, Zhe Chen, Baosheng Yu, and Dacheng Tao. Contrastive boundary learning for point cloud segmentation. In *CVPR*, 2022. S1, S7

[7] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021. S1, S2, S7, S9

[8] 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 *ICCV*, 2019. S2, S6, S7, S8

[9] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *CVPR*, 2018. S6, S7, S8

[10] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *ICRA*, 2019. S6

[11] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *IROS*, 2019. S6

[12] Iñigo Alonso, Luis Riazuelo, Luis Montesano, and Ana C Murillo. 3d-mininet: Learning a 2d representation from point clouds for fast and efficient 3d lidar semantic segmentation. In *IROS*, 2020. S6

[13] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *CVPR*, 2020. S6

[14] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *CVPR*, 2020. S6

[15] 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 *CVPR*, 2020. S6

[16] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In *ECCV*, 2020. S6

[17] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: fast, uncertainty-aware semantic segmentation of lidar point clouds for autonomous driving. *arXiv preprint arXiv:2003.03653*, 2020. S6

[18] Feihu Zhang, Jin Fang, Benjamin Wah, and Philip Torr. Deep fusionnet for point cloud semantic segmentation. In *ECCV*, 2020. S6

[19] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In *AAAI*, 2021. S6

[20] Yuenan Hou, Xinge Zhu, Yuexin Ma, Chen Change Loy, and Yikang Li. Point-to-voxel knowledge distillation for lidar semantic segmentation. In *CVPR*, 2022. S6

[21] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *ECCV*, 2020. S6

[22] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017. S7

[23] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *3DV*, 2017. S7

[24] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *NeurIPS*, 2018. S7

[25] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, 2018. S7

[26] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. In *CVPR*, 2018. S7

[27] Li Jiang, Hengshuang Zhao, Shu Liu, Xiaoyong Shen, Chi-Wing Fu, and Jiaya Jia. Hierarchical point-edge interaction network for point cloud semantic segmentation. In *ICCV*, 2019. S7

[28] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *CVPR*, 2019. S7

[29] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. Pointweb: Enhancing local neighborhood features for point cloud processing. In *CVPR*, 2019. S7

[30] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 2019. S7

[31] Siqi Fan, Qiulei Dong, Fenghua Zhu, Yisheng Lv, Peijun Ye, and Fei-Yue Wang. Scf-net: Learning spatial contextual features for large-scale point cloud segmentation. In *CVPR*, 2021. S7

[32] Shi Qiu, Saeed Anwar, and Nick Barnes. Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion. In *CVPR*, 2021. S7

[33] Tao Lu, Limin Wang, and Gangshan Wu. Cga-net: Category guided aggregation for point cloud semantic segmentation. In *CVPR*, 2021. S7

[34] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified trans-

former for 3d point cloud segmentation. In *CVPR*, 2022. S7
[35] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In *NeurIPS*, 2022. S7
[36] Fabian Duerr, Mario Pfaller, Hendrik Weigel, and Jürgen Beyerer. Lidar-based recurrent 3d semantic segmentation with temporal memory alignment. In *3DV*, 2020. S8
[37] Hanyu Shi, Guosheng Lin, Hao Wang, Tzu-Yi Hung, and Zhenhua Wang. Spsequencenet: Semantic segmentation network on 4d point clouds. In *CVPR*, 2020. S8