CPCM: Contextual Point Cloud Modeling for Weakly-supervised Point Cloud Semantic Segmentation (Supplementary Materials)

Lizhao Liu[1,2] Zhuangwei Zhuang[1,2] Shangxin Huang[1] Xunlong Xiao[1] Tianhang Xiang[1]
Cen Chen[1] Jingdong Wang[3] Mingkui Tan[1,2†]
[1]South China University of Technology [2]Pazhou Lab [3]Baidu Inc.
{selizhaoliu, z.zhuangwei, sevtars, sexxl, sexiangtianhang}@mail.scut.edu.cn,
{chencen, mingkuitan}@scut.edu.cn, wangjingdong@baidu.com

We organize our supplementary materials as follows:

- In Section A, we provide more experiment details and results on pilot studies that inspect the contextual comprehension ability learned by the consis-based baseline and the proposed Contextual Point Cloud Modeling (CPCM) method.
- In Section B, we introduce the details of three losses, namely, the supervised cross-entropy loss $\mathcal{L}_{seg}$, the consistency loss $\mathcal{L}_{consis}$ and the proposed masked consistency loss $\mathcal{L}_{mask}$, in the objective of CPCM.
- In Section C, we present the technical details of data augmentation for the point cloud data.
- In Section D, we study the effect of hyper-parameters α and β that control the optimization strength on the consistency loss and the mask consistency loss, respectively.
- In Section E, we conduct ablation studies on the masked consistency loss $\mathcal{L}_{mask}$.
- In Section F, we supply experiments on the outdoor dataset SemanticKITTI [2].
- In Section G, we conduct further experiments on the transformer architecture PTv2 [6].
- In Section H, we compare the performance of the proposed CPCM with unsupervised pre-training methods.
- In Section I, we give more implementation details on producing a strong consistency-based baseline.
- In Section J, we provide more visualization results on ScanNet V2 and S3DIS.
- In Section K, we analyze the failure case of the proposed CPCM.

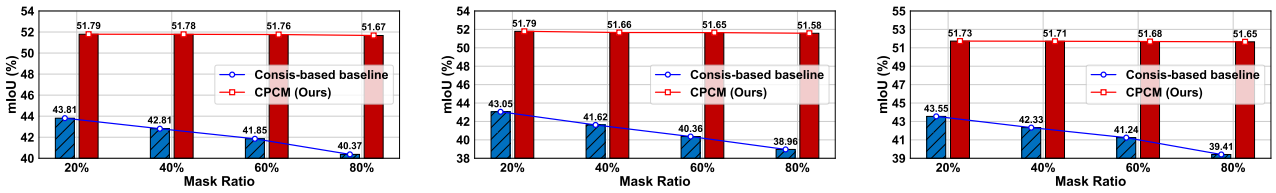
A. More Results on Pilot Studies

In this section, we inspect the context comprehension ability learned by the consis-based baseline and the proposed CPCM by a series of pilot studies. To this end, we conduct experiments with the model trained by the proposed CPCM and the consis-based training methods [5, 9]. To be specific, we train the model on two datasets: ScanNet V2 [3] with 0.01% annotations and S3DIS [1] with 0.01% annotations. Then, we design a masked evaluation protocol introduced below to quantitatively and qualitatively analyze each model's context comprehension ability.

Masked evaluation. We require the model to perform segmentation given a masked point cloud as input. In this sense, the masked part serves as *context-to-be-filled* and the model shall understand the masked parts' surroundings, aka contextual information, for accurate segmentation. The masked point cloud is obtained by three masking strategies detailed below.

Masking strategies. We introduce three masking strategies: partial-instance masking and complete instance masking, which evaluate the context comprehension within the instance and region-wise masking, which evaluates the context understanding across instances. To be specific, **1) Partial-instance masking** randomly masks some RGB features within each instance. **2) Region-wise masking** divides the point cloud into a set of regions and masks all RGB features of the randomly selected regions. **3) Complete-instance masking** completely erases the RGB features of the randomly selected instances. *Note that we use the instance annotation on ScanNet V2 and S3DIS for masked evaluation only and no instance annotation is used for model training.* During the evaluation, for three kinds of masking strategies, we gradually increase the mask ratio to increase the difficulty of the context comprehension task.

Results. The mIoU results w.r.t. different mask ratios are shown in Figures I and III for ScanNet V2 and S3DIS, respectively. As the mask ratio increases, we observe that our CPCM slightly decreases $0.08\% \sim 0.21\%$ on ScanNet V2 and $4.11\% \sim 5.15\%$ on S3DIS, while the consis-based baseline considerably drops $3.44\% \sim 4.14\%$ on ScanNet V2 and $10.37\% \sim 12.26\%$ on S3DIS. We also present the visual comparison results in Figures II and IV for ScanNet V2 and S3DIS, respectively. Note that we select the visual results under mask ratio 40% for better visualization. We can see that our CPCM performs well under both standard and masked evaluations while the consis-based baseline fails to fill the masked part. These results demonstrate that the proposed CPCM has much stronger context comprehension ability over the consis-based baseline and achieves better and more robust point cloud semantic segmentation.



(a) Partial-instance masked evaluation results. (b) Region-wise masked evaluation results. (c) Complete-instance masked evaluation results.

Figure I: Masked evaluation results on ScanNet V2 [3] to inspect the contextual perception ability.

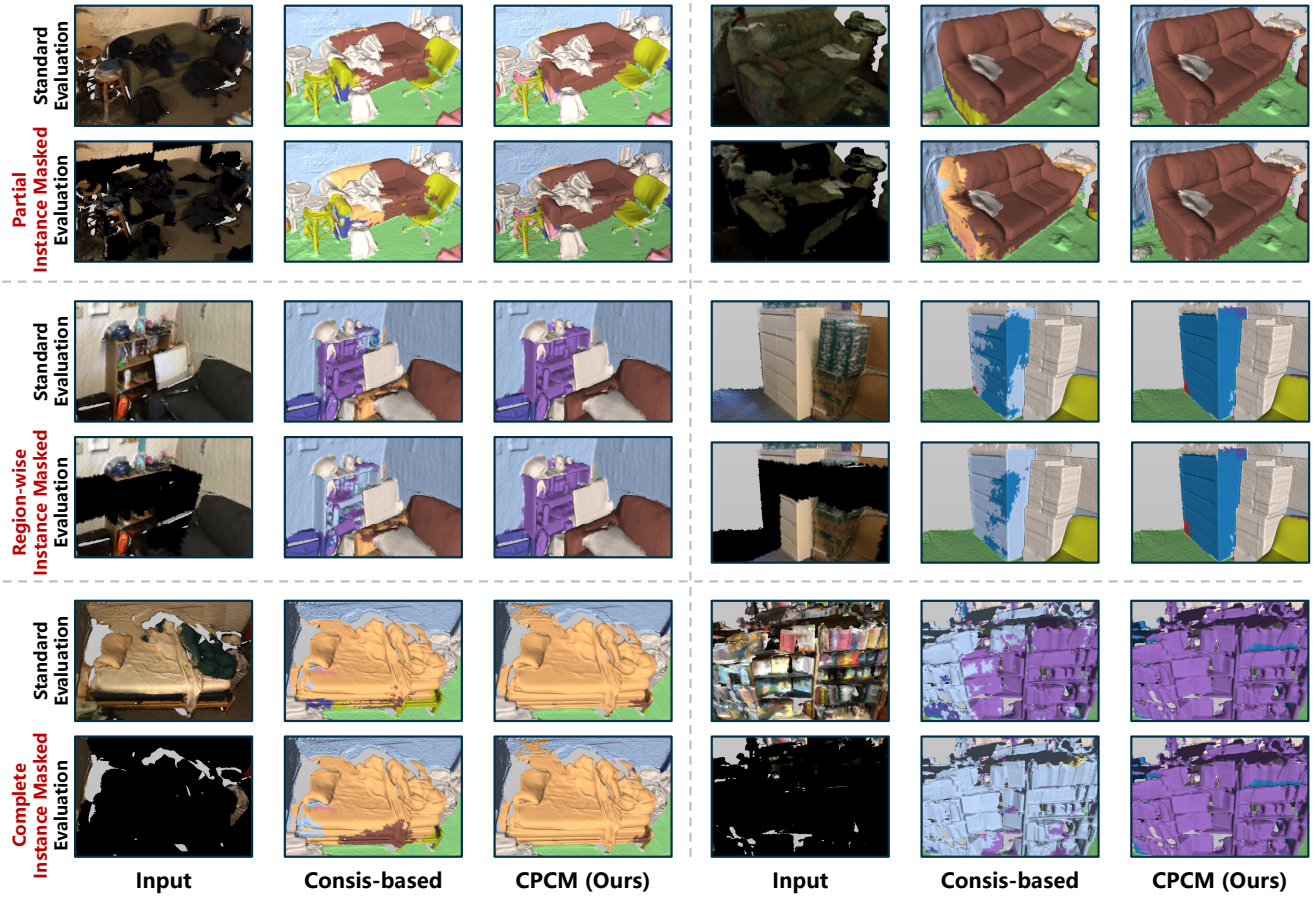
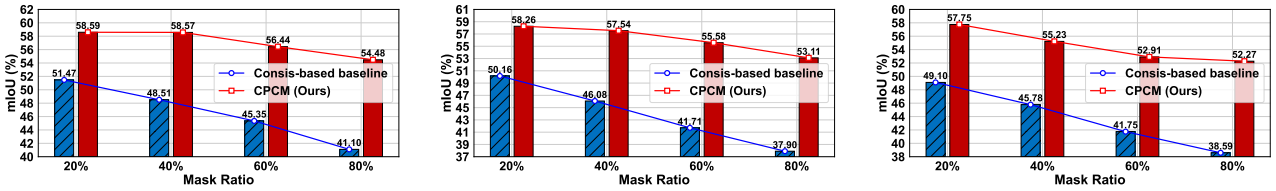


Figure II: Visual comparison of results from different methods on ScanNet V2 (mask ratio is 40%).



(a) Partial-instance masked evaluation results. (b) Region-wise masked evaluation results. (c) Complete-instance masked evaluation results.

Figure III: Masked evaluation results on S3DIS [1] to inspect the contextual perception ability.

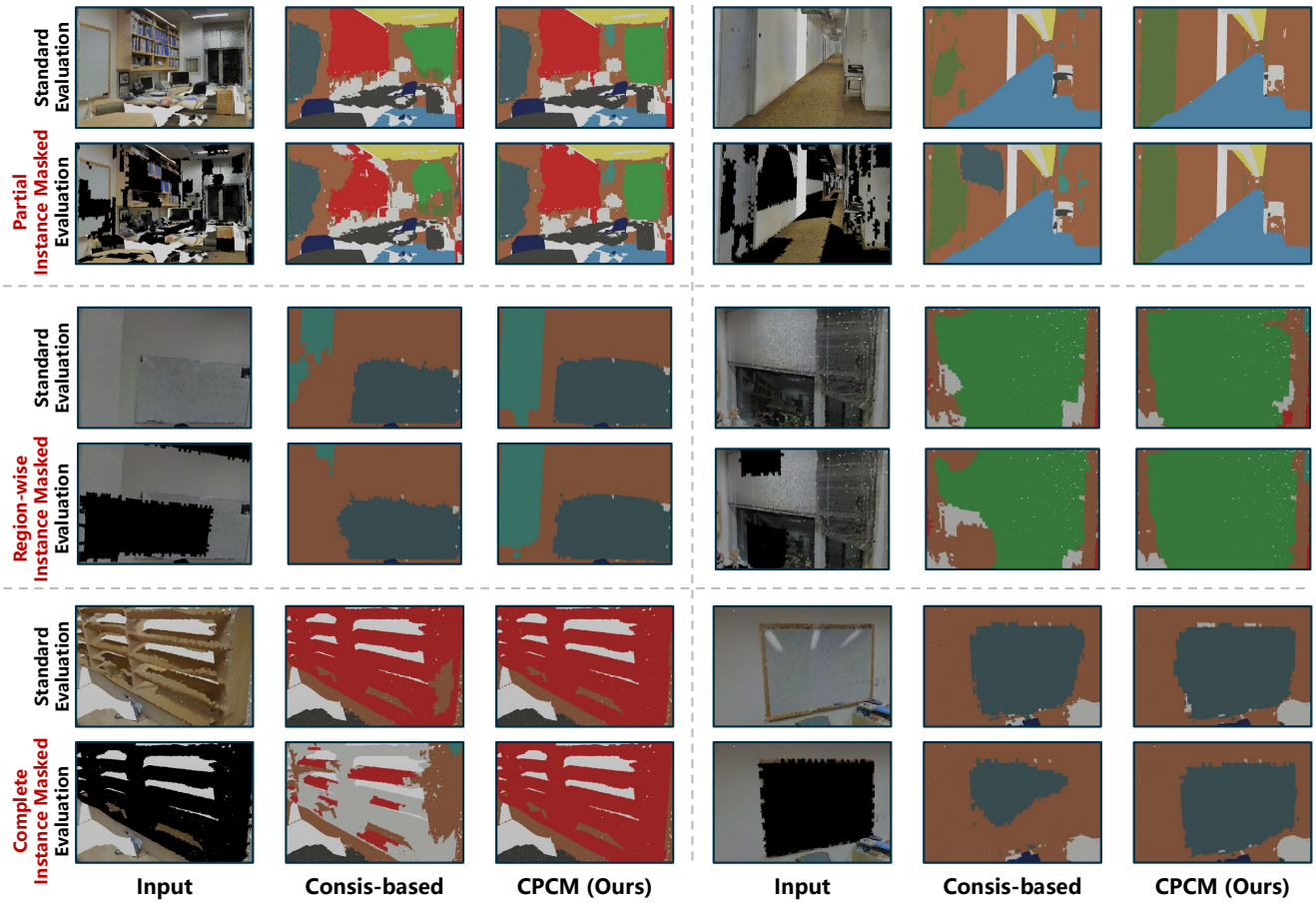


Figure IV: Visual comparison of results from different methods on S3DIS (mask ratio is 40%).

B. Detailed Formulations of the Loss Functions

For convenience, we show the overall scheme of the proposed CPCM in Figure V. Recall that in Section 3.3 of the main paper, we have introduced the objective of CPCM. Specifically, we propose to learn the masked feature consistency to improve the context comprehension ability of the model. Following the consis-based methods [5, 9], we use the supervised cross-entropy loss on the labeled points and the JS-divergence consistency loss on features from different augmentations. Therefore, the overall objective of the proposed CPCM is defined as

$$\mathcal{L}_{\text{CPCM}} = \mathcal{L}_{seg} + \alpha\mathcal{L}_{consis} + \beta\mathcal{L}_{mask}, \tag{I}$$

where $\mathcal{L}_{seg}, \mathcal{L}_{consis}, \mathcal{L}_{mask}$ indicate the cross-entropy loss, consistency loss and masked consistency loss, respectively. Here, α and β are hyper-parameters that control the optimization strength to learn feature consistency across augmentations and contextual information, respectively.

In this section, we provide detailed formulations of the used loss functions. Given differently augmented point clouds $\mathbf{P}_1, \mathbf{P}_2$ from $\mathbf{P}$, the sparse label $\mathbf{Y}$, and the labeled index set $\mathcal{S}$, we first obtain the point-wise classification logits by $\mathbf{Z}_1 = \text{Softmax}\big(f_\theta(\mathbf{P}_1)\big)$ and $\mathbf{Z}_2 = \text{Softmax}\big(f_\theta(\mathbf{P}_2)\big)$.

Cross-entropy loss. The supervised cross-entropy loss $\mathcal{L}_{seg}$ is computed as follows:

$$\mathcal{L}_{seg} = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}} CE(\mathbf{Z}_1[s], \mathbf{Y}[s]) + CE(\mathbf{Z}_2[s], \mathbf{Y}[s]) = -\frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}} \log\Big(\mathbf{Z}_1[s]\,[\mathbf{Y}[s]]\Big) + \log\Big(\mathbf{Z}_2[s]\,[\mathbf{Y}[s]]\Big). \tag{II}$$

Consistency loss. The consistency loss $\mathcal{L}_{consis}$ is calculated over the unlabeled (or all) points as follows:

$$\mathcal{L}_{consis} = \frac{1}{N}\sum_{n} JS(\mathbf{Z}_1[n], \mathbf{Z}_2[n]) = -\frac{1}{N}\sum_{n} \mathbf{Z}_1[n]\log\Big(\frac{\mathbf{Z}^{'}[n]}{\mathbf{Z}_1[n]}\Big) + \mathbf{Z}_2[n]\log\Big(\frac{\mathbf{Z}^{'}[n]}{\mathbf{Z}_2[n]}\Big), \tag{III}$$

$$\mathbf{Z}^{'} = (\mathbf{Z}_1 + \mathbf{Z}_2)/2. \tag{IV}$$

Masked consistency loss. Last, given the masked point cloud $\mathbf{P}_m$, we compute its features computed by $\mathbf{Z}_m = \text{Softmax}\big(f_\theta(\mathbf{P}_m)\big)$ and derive our masked consistency loss $\mathcal{L}_{mask}$ as follows:

$$\mathcal{L}_{mask} = \frac{1}{N}\sum_{n} JS(\mathbf{Z}_1^{'}[n], \mathbf{Z}_m[n]) + JS(\mathbf{Z}_2^{'}[n], \mathbf{Z}_m[n]), \tag{V}$$

where we stop the gradient flow for the "ground truth" unmasked features $\mathbf{Z}_1, \mathbf{Z}_2$ by the `Detach` operation in PyTorch,

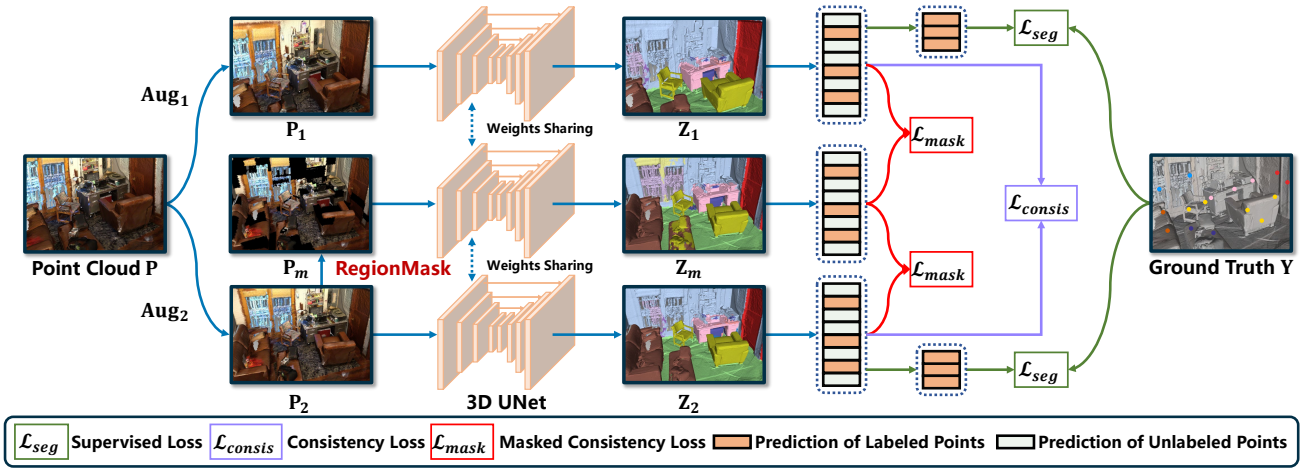


Figure V: Overall scheme of our CPCM method. Given a point cloud $\mathbf{P}$, we first apply two random augmentations and our region-wise masking to obtain the augmented point clouds $\mathbf{P}_1, \mathbf{P}_2$ and the masked point cloud $\mathbf{P}_m$, respectively. Then, the features $\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_m$ are extracted by a weight-sharing 3D UNet. The supervised cross-entropy loss $\mathcal{L}_{seg}$ is computed over labeled features and a consistency loss $\mathcal{L}_{consis}$ is computed on $\mathbf{Z}_1, \mathbf{Z}_2$. Last, our masked consistency loss $\mathcal{L}_{mask}$ enforces the feature consistency between $\mathbf{Z}_1, \mathbf{Z}_m$ and $\mathbf{Z}_2, \mathbf{Z}_m$ to help the model focus on learning contextual information.

C. Technical details on the data augmentation

In this section, we detail the data augmentation used on the point cloud data. We apply the same data augmentation method to get $\mathbf{P}_1$ and $\mathbf{P}_2$. Following previous methods [4, 7], we use data augmentations including: `RandomDropOut`, `RandomHorizontalFilp`, `ColorAutoContrast`, `ColorTranslation` and `ColorJitter`. The data augmentation is called two times to create differently augmented point clouds to learn feature consistency. Since the augmentation method is the same for $\mathbf{P}_1$ and $\mathbf{P}_2$, using any one of them to get $\mathbf{P}_m$ is feasible.

D. Effect of hyper-parameters α and β

In this section, we investigate the effect of hyper-parameters α and β that control the optimization strength on consistency loss and mask consistency loss, respectively. We use S3DIS for fast evaluation considering that the number of the training sample is relatively small on S3DIS (204 on S3DIS *vs.* 1201 on ScanNet V2). Moreover, the optimal hyper-parameters for different annotation ratios may vary. Thus, we conduct experiments on S3DIS with annotation ratios 0.01% and 0.1%. We present the experiment results in Table I and our analysis is as follows.

Effect of hyper-parameter α. The role of α is to control the learning from unlabeled data. As shown in Table I, the optimal value of α is 5 and 1 for the extreme-limited and the limited annotation settings, respectively. Moreover, we observe that the performance does not improve when $\alpha > 1$ under the 0.1% setting. The results indicate that the consistency loss is useful for learning representation as the annotation ratio decreases.

Effect of hyper-parameter β. The hyper-parameter β is to control the learning of the masked features prediction task based on unmasked surroundings, which helps the model harness the contextual information in a scene. To better explore the effect of the masked consistency loss, we simply set $\alpha = 0$, which does not apply the consistency loss to learn the weakly-supervised segmentation model. As shown in Table I, the optimal value is 10 and 5 for the extreme-limited and limited annotation setting, which is larger than the optimal value of α *i.e.,* $(\beta, \alpha) = (10, 5)$ when the 0.01% setting and $(\beta, \alpha) = (5, 1)$ for the 0.1% setting. The larger optimal value of β indicates that learning contextual information is more effective than learning feature consistency across augmentations. Last, without the consistency loss, the optimal performance of CPCM beat the consis-based method considerably, showing the advantage of considering point contextual relation over point-wise consistency across augmentations only.

Based on the above results, for both S3DIS and ScanNet V2, we simply set (α, β) to $(5, 10)$ and $(1, 5)$ for annotation ratio $< 0.1\%$ and $\geq 0.1\%$, respectively. We admit there may be more optimal hyper-parameters by tuning under the specific dataset and annotation settings as well as tuning (α, β) simultaneously. For the sake of simplicity, we decide to apply the above coarse settings throughout our experiments.

	S3DIS 0.01%				S3DIS 0.1%			
α	1	2	5	10	1	2	5	10
CPCM ($\beta = 0$)	48.6	50.7	**52.9**	51.8	**65.0**	64.9	64.7	64.3
β	1	2	5	10	1	2	5	10
CPCM ($\alpha = 0$)	51.8	53.5	56.9	**59.2**	65.2	65.6	**66.3**	64.0

Table I: Ablation studies on hyper-parameters α for the Consis-based method and β for the proposed CPCM.

E. Ablation studies on the masked consistency loss $\mathcal{L}_{mask}$

In this section, we provide ablation studies on masked consistency loss. To compute $\mathcal{L}_{mask}$ for $\mathbf{Z}_1$ and $\mathbf{Z}_m$, we align them before the loss calculation. For more details, refer to Section I. As mentioned in Section C, both $\mathbf{P}_1$ and $\mathbf{P}_2$ are the "unmasked" version of $\mathbf{P}_m$. Thus, minimizing the distribution gap between both $\mathbf{Z}_1, \mathbf{Z}_m$ and $\mathbf{Z}_2, \mathbf{Z}_m$ is helpful to learn contextual information. We conduct experiments on the S3DIS dataset with 0.01% annotation. As shown in Table II, using both $JS(\mathbf{Z}_1, \mathbf{Z}_m)$ and $JS(\mathbf{Z}_2, \mathbf{Z}_m)$ achieves the best result.

$\mathcal{L}_{mask}$		mIoU (%)
$JS(\mathbf{Z}_1, \mathbf{Z}_m)$	$JS(\mathbf{Z}_2, \mathbf{Z}_m)$	
✘	✘	47.7
✔	✘	56.5 (+8.8)
✘	✔	57.2 (+9.5)
✔	✔	**59.3** (+11.6)

Table II: Ablation studies of $\mathcal{L}_{mask}$ on S3DIS.

F. Further experiments on the outdoor dataset

To further demonstrate the performance of our CPCM, we provide the quantitative results on the outdoor dataset, SemanticKITTI [2]. Since we followed previous works [4, 7, 8] to use MinkowskiEngine to implement our CPCM, we conduct experiments on the front view part of the SemanticKITTI that provides both XYZ and RGB features for convenience. As shown in Table III, our CPCM consistently provides improvement over the MinkNet and the consis-based baseline. Moreover, thanks to the strong contextual modeling ability, CPCM surpasses the baselines with more annotation, *e.g.,* CPCM (**44.0**, **0.1%**) > consis-based (43.7, 1%) > MinkNet (37.0, 1%). These results demonstrate that our CPCM is able to perform well not only in indoor but also in outdoor scenarios.

Method	Setting		
	1%	0.1%	0.01%
MinkNet	37.0	30.8	23.7
Consis-based baseline	43.7 (+6.7)	38.8 (+8.0)	30.0 (+6.3)
CPCM (Ours)	**47.8** (+10.8)	**44.0** (+13.2)	**34.7** (+11.0)

Table III: Results of mIoU (%) on SemanticKITTI. For reference, the mIoU for fully-supervised MinkNet is 56.4%.

G. Further experiments on the transformer

To investigate the effectiveness of our CPCM on transformer architecture, we apply CPCM on PTv2 [6], a transformer-based architecture trained in a fully-supervised manner. Since the transformer is generally more data-hungry, we conduct experiments on less weakly-supervised settings (10% or fully supervised) and compare to PTv2. To be specific, we substitute the backbone of CPCM, *i.e.,* MinkNet to PTv2 and the results are shown in Table IV. We observe that our CPCM improves the performance of PTv2 by **5.7%** and **1.6%** with 10% and 100% annotations, respectively. These results verify the effectiveness of CPCM in transformer architecture.

Settings	PTv2	PTv2†	PTv2 + CPCM (Ours)
Fully	71.6	69.1	**70.7** (+1.6)
10%	-	54.6	**60.3** (+5.7)

Table IV: Comparisons with PTv2 on S3DIS, where † denotes the results of our implementation.

H. Comparison with Unsupervised Pre-training Methods

Unsupervised point cloud pre-training methods can learn useful representations from mass unlabeled data. Thus, existing unsupervised pre-training methods [4, 7] use the pretrained model as initialization and finetune the model on the downstream weakly-supervised point cloud segmentation task. In this section, we investigate the potential of the proposed CPCM by challenging the strong and universal unsupervised pre-training methods: Point Contrast (PC) [7] and Contrastive Scene Context (CSC) [4]. They both leverage point-wise contrastive learning to pre-train the segmentation network. We admit that there are many works on unsupervised point cloud pre-training topics. Since PC and CSC have been evaluated under the weakly-supervised setting, we choose them as our baselines. The results are shown in Table V. The proposed CPCM outperforms PC and CSC under various annotation settings, often by a large margin. Specifically, our CPCM outperforms CSC by 8.87% under the extreme-limited annotation setting 20pts. Moreover, our CPCM train by 50 pts and 100 pts are able to surpass the CSC trained by 100 pts and 200 pts, respectively. Note that our CPCM does not require the pre-training phases and is more suitable for downstream scenarios without large pre-training datasets and computation power.

Method	20 pts	50 pts	100 pts	200 pts
PC [7]	N/A	N/A	N/A	67.80
CSC [4]	53.60	60.70	65.70	68.20
CSC* [4]	53.80	62.90	66.90	69.00
CPCM (Ours)	**62.67** (+8.87)	**67.89** (+4.99)	**69.67** (+2.77)	**70.32** (+1.32)

Table V: Comparisons with unsupervised pre-training methods on the ScanNet V2 limited annotated points (pts) per-scene benchmark. * indicates using the active scheme to label representative points.

I. More Implementation Details

In this section, to facilitate further research, we provide two important implementation details that affect the performance considerably: the point alignment operation and tuning the hyper-parameter weight decay.

Point alignment. Note that for both the consistency loss and the masked consistency loss, the alignment operation is required to align two scenes' points before the loss calculation. This is because different augmentations such as random point dropout, geometric clipping and point voxelization would drop some points, which leads to points' misalignment for the two stream data flow. We can resolve this issue by two means: **1) Input level alignment**: all points are aligned *before* feeding into the segmentation network, which leads to the more sparse point cloud data and the loss of some valuable annotations. **2) Feature level alignment**: all points are aligned *after* the feature extraction stage, which causes some feature inconsistency but resolves all issues incurred by the input level alignment solution. We implement both solutions to find out which is better and the results are shown in Table VI. We observe that the feature level alignment strategy performs better w.r.t. different annotation settings and datasets. We attribute the success of feature-level alignment to 1) training and testing under the same input distribution and 2) retraining valuable annotations. Therefore, we choose the feature level alignment as our default point alignment strategy throughout experiments.

Alignment Strategy	ScanNet V2		S3DIS	
	0.01%	0.1%	0.01%	0.1%
Input Level	39.9	59.1	46.9	59.6
Feature Level	**44.2**	**61.8**	**52.9**	**64.9**

Table VI: Effect of the point alignment position input level *vs.* feature level on the consis-based method.

Weight decay. Due to the limited annotation nature in weakly-supervised point cloud segmentation, overfitting is an issue we should consider properly. Thus, we carry out a simple but straightforward way to alleviate the overfitting issue: tuning the weight decay to control the regularization intensity. The results are shown in Table VII. As weight decay increases from $1e^{-4}$ to $1e^{-3}$, both MinkNet and consis-based method achieves better results, which indicates that higher weight decay is able to alleviate the overfitting issue. However, a large weight decay, *i.e.,* $1e^{-2}$ causes the underfitting issue, especially on ScanNet V2 that with thousands of scene point cloud data to be fitted. Thus, we set weight decay to $1e^{-3}$ as our default choice.

Weight Decay	ScanNet V2 0.01%		S3DIS 0.01%	
	MinkNet	Consis-based	MinkNet	Consis-based
$1e^{-4}$	36.7	41.3	45.9	50.9
$1e^{-3}$	**37.6**	**44.2**	**47.7**	**52.9**
$1e^{-2}$	15.5	14.0	45.1	50.5

Table VII: Effect of weight decay on the two baselines: MinkNet and consis-based method.

J. More Qualitative Results

In this section, we demonstrate the advantage of CPCM with more visualization results in Figure VI. We summed up CPCM's advantage as follows:

Better at distinguishing adjacent objects. CPCM is able to disguise geometrically close and appearance similar objects, as shown in row 1 of ScanNet V2 (curtain and wall) and row 4 of S3DIS (door and wall).

Better at covering the whole object. CPCM does well in covering large objects as shown in rows 2,4 of ScanNet V2 (bed and table) and rows 1,3 of S3IDS (board and ceiling), indicating CPCM's long-range context comprehension ability.

Better at recognizing the object with complex structures. CPCM performs reasonably well at recognizing objects with complex geometric structures and appearance as shown in row 3 of ScanNet V2 (curtain) and row 2 of S3DIS (bookcase).

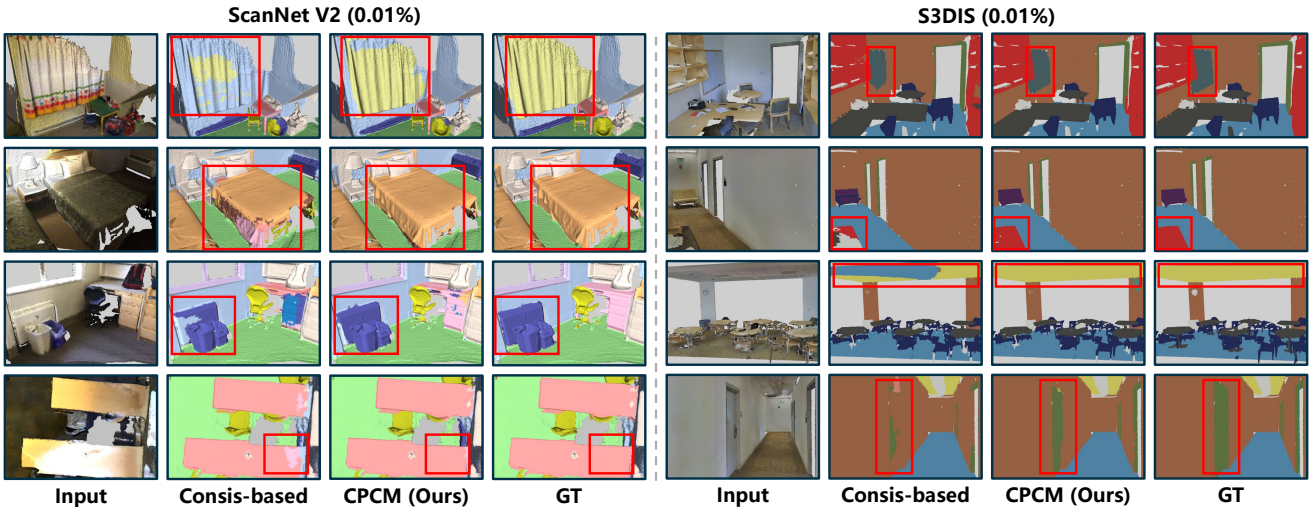


Figure VI: More qualitative comparison between the consis-based method and our CPCM on the ScanNet V2 and S3DIS. We highlight the prediction difference between consis-based method and our CPCM with a red box.

K. Failure case analysis of CPCM

Our CPCM may fail to effectively distinguish similar classes in the point cloud with a large part of the missing region. Since CPCM heavily relies on the *complete* context to perform accurate segmentation, point clouds with lots of *missing* regions may not provide sufficient context to make correct predictions. For example, in Figure VII, CPCM, unfortunately, hallucinates the door as the window.

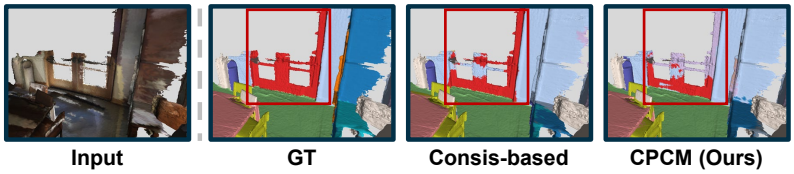


Figure VII: Failure cases of our CPCM on the ScanNet V2.

References

[1] 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*, pages 1534–1543, 2016. 2, 4

[2] 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 *CVPR*, pages 9297–9307, 2019. 1, 7

[3] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017. 2, 3

[4] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *CVPR*, pages 15587–15597, 2021. 6, 7, 8

[5] Mengtian Li, Yuan Xie, Yunhang Shen, Bo Ke, Ruizhi Qiao, Bo Ren, Shaohui Lin, and Lizhuang Ma. Hybridcr: Weakly-supervised 3d point cloud semantic segmentation via hybrid contrastive regularization. In *CVPR*, pages 14930–14939, 2022. 2, 5

[6] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *NeurIPS*, 35:33330–33342, 2022. 1, 7

[7] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *ECCV*, pages 574–591, 2020. 6, 7, 8

[8] Cheng-Kun Yang, Ji-Jia Wu, Kai-Syun Chen, Yung-Yu Chuang, and Yen-Yu Lin. An mil-derived transformer for weakly supervised point cloud segmentation. In *CVPR*, pages 11830–11839, 2022. 7

[9] Yachao Zhang, Yanyun Qu, Yuan Xie, Zonghao Li, Shanshan Zheng, and Cuihua Li. Perturbed self-distillation: Weakly supervised large-scale point cloud semantic segmentation. In *ICCV*, pages 15520–15528, 2021. 2, 5