Supplementary Material for MM-3DScene

Appendix A: Visualization Results

Fig. c visualizes the masked reconstruction results of MM-3DScene. It can be observed that: **i)** For masked **input**, our mask strategy *preserves* Informative Points to provide basic geometric information, which explicitly reduces the ambiguity during masked reconstruction. **ii)** For **target**, instead of being the original intact scene, it is a relatively more complete one with a smaller masking ratio. This prompts models to *concentrate* on reconstructing the local regional 3D structures where models focus on recovering *regional* geometric patterns. **iii)** For reconstruction **result**, our model is able to recover the masked areas, suggesting it successfully learned numerous visual representations. For example, our method works well to recover details of the masked foreground objects (*e.g*. table and chair). For the background surfaces (*e.g*. floor, wall), our method can also achieve a smooth and complete recovery. In addition to the visualization of the pre-training reconstruction results, as shown in the Fig. a, we present the visualization results of the downstream semantic segmentation task. It can be seen that compared with other methods, our method can correct the results in some areas where the prediction is inaccurate.

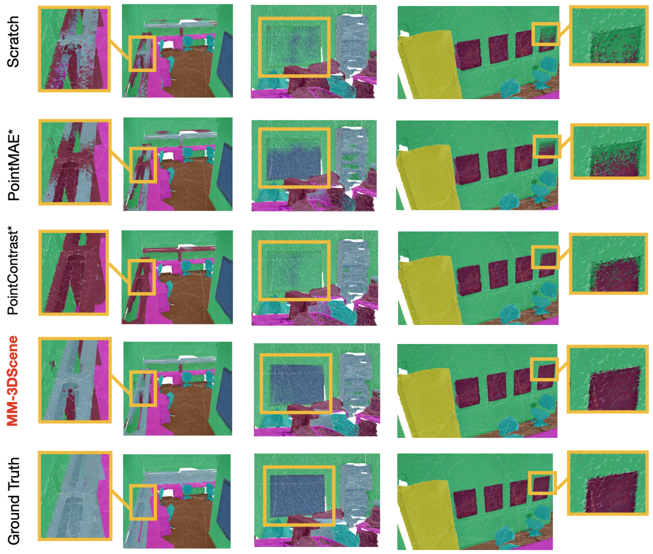

Figure a. Qualitative results on S3DIS semantic segmentation.

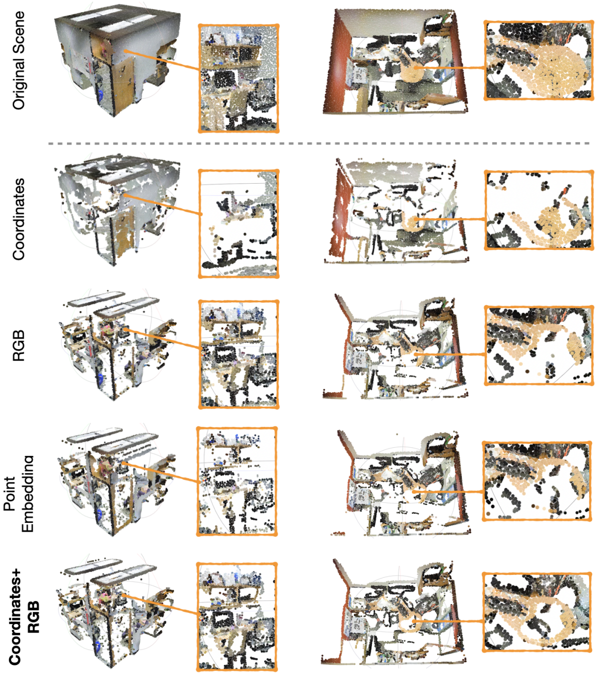

Figure b. Visualization of masked scene guided by different formats of local statistics. In this figure, the **contour of a table** (*i.e*., the representative geometric structures) is accurately found and preserved, when calculating the local difference of coordinates+RGB.

Appendix B: More Ablations of Masked Reconstruction

More formats of local statistics. In our Local-Statistics-Guided Masking, we exploit local statistics to discover informative points, aiming to *accurately* retain the representative geometric structures. The local statistics are denoted by the local difference between each point and its neighboring points in terms of coordinates and colors. Here we investigate other possible formats of the local difference, including point embedding difference (gained by applying MLPs [2] on each point), only coordinates difference, and only RGB difference. Fig. b shows that considering the local difference of both coordinates and RGB is the most applicable way to find informative points. For example in this figure, the *structural contour* of a table is *accurately* found and

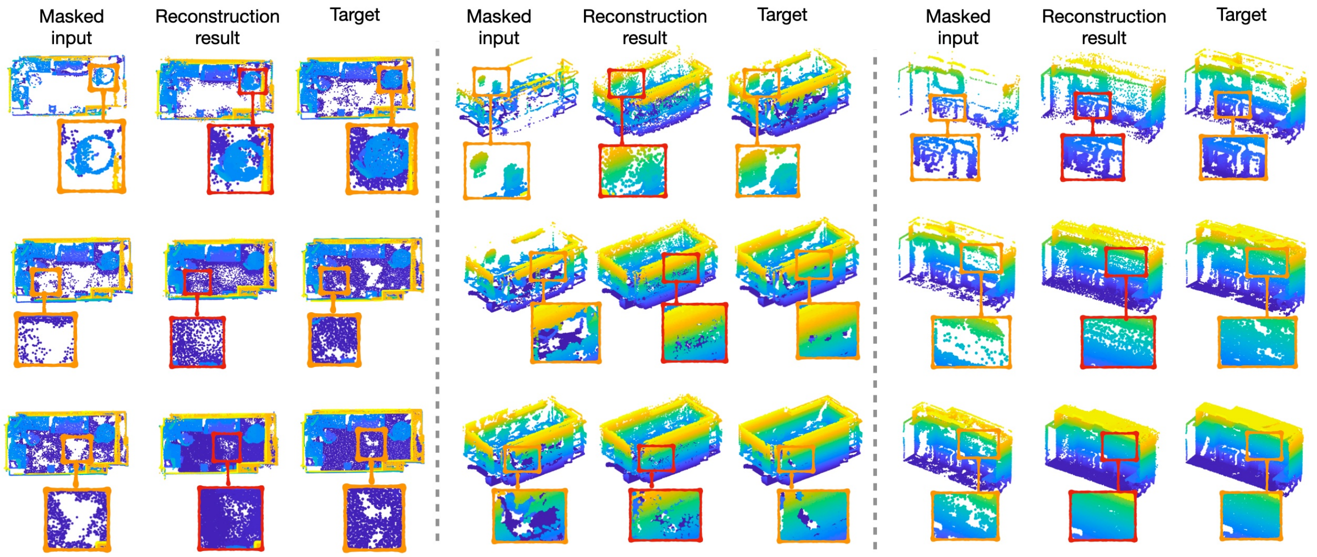

Figure c. Qualitative results of MM-3DScene masked reconstruction pretext task. Our method can effectively reconstruct the masked areas, suggesting it learned rich visual representations for understanding 3D scenes.

	Pre-training method	Local Statistic	mIoU	mAcc
i	Scratch	-	70.4	76.5
ii	MM-3DScene (w/o L_{CSD})	Coordinates	70.6 (+0.2	76.5 (+0.0)
iii	MM-3DScene (w/o L_{CSD})	RGB	70.9 (+0.5)	77.0 (+0.5)
iv	MM-3DScene (w/o L_{CSD})	point embedding	70.9 (+0.5)	76.9 (+0.4)
v	MM-3DScene (w/o L_{CSD})	**Coordinates + RGB**	**71.1** (+0.7)	**77.2** (+0.7)

Table a. MM-3DScene (w/o L_{CSD}) guided by **different formats of local statistics** for S3DIS semantic segmentation.

preserved, when guided by the local differences of coordinates+RGB, which provides useful information hints for recovering the masked interior area of this table. As a result, our method performs best when the masking is guided by the local statistic of coordinates+RGB, as listed in Table a.

Ablation studies of reconstruction gap. Our method uses the incremental masking ratio $\theta = \{\theta_1, ..., \theta_t, ..., \theta_T\}$ to progressively mask the scene. During the masked reconstruction, the masking ratio is θ_t for the input scene, and $\theta_{t-\eta}$ for the target scene, where η indicates the masked gap to be recovered and latently influences the *difficulty* of the pretext task. Fig. d provides the ablation study of such reconstruction gap, where our model enjoys the least difficulty and performs best under $\theta_t - \theta_{t-\eta} = 0.1$, and degrades when the gap becomes larger. Additionally, we also implement the random reconstruction gap, which probably causes more ambiguity, yielding 70.36% mIoU.

Appendix C: Other Backbones with MM-3DScene

In the main paper, we adopt VoteNet [6] as the backbone for object detection, and Point Transformer [16] for seman-

Method	Model Size	Train Time	Infer Time	mAP@0.25	mAP@0.5
VoteNet [6] (scratch)	0.95M	5.9h	0.2s	58.7	35.4
MM3DScene + VoteNet	1.48M	15.6h	-	63.1 (+4.4)	41.5 (+6.1)
H3DNet [14] (scratch)	4.74M	12.3h	7.3s	64.8	47.4
MM3DScene + H3DNet	6.87M	36.1h	-	66.8 (+2.0)	48.9 (+1.5)

Table b. 3D object detection results on ScanNetv2. The baseline results come from official code implementations. The training and inference times are evaluated with the same training settings.

Method	Model Size	Train Time	Infer Time	mIoU
PointTrans [16] (scratch)	7.76M	17.3h	4.36s	70.4
MM3DScene + PointTrans	8.63M	29.1h	-	71.9 (+1.5)
Stratified Trans* [3] (scratch)	8.02M	45.7h	11.77s	70.3
MM-3DScene + Stratified Trans*	8.89M	73.2h	-	71.6(+1.3)

Table c. 3D semantic segmentation results on S3DIS. The baseline results come from official code implementations. The training and inference times are evaluated with the same training settings.

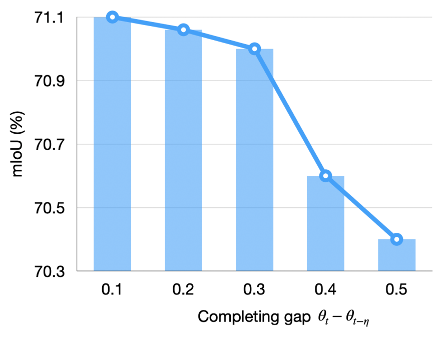

Figure d. Ablation study of **masked reconstruction gap** on S3DIS semantic segmentation (based on MM-3DScene w/o L_{CSD}).

Method	mIoU	ceil.	floor	wall	beam	col.	win.	door	table	chair	sofa	bookc.	board	clu.
PointNet [7]	41.1	88.8	97.3	69.8	0.1	3.9	46.3	10.8	58.9	52.6	5.9	40.3	26.4	33.2
SegCloud [9]	48.9	90.1	96.1	69.9	0.0	18.4	38.4	23.1	70.4	75.9	40.9	58.4	13.0	41.6
TangentConv [8]	52.6	90.5	97.7	74.0	0.0	20.7	39.0	31.3	77.5	69.4	57.3	38.5	48.8	39.8
SPGraph [4]	58.0	89.4	96.9	78.1	0.0	42.8	48.9	61.6	84.7	75.4	69.8	52.6	2.1	52.2
PCNN [1]	58.3	92.3	96.2	75.9	0.3	6.0	69.5	63.5	65.6	66.9	68.9	47.3	59.1	46.2
RNNFusion [12]	57.3	92.3	98.2	79.4	0.0	17.6	22.8	62.1	80.6	74.4	66.7	31.7	62.1	56.7
Eff 3D Conv [13]	51.8	79.8	93.9	69.0	0.2	28.3	38.5	48.3	73.6	71.1	59.2	48.7	29.3	33.1
PointCNN [5]	57.3	92.3	98.2	79.4	0.0	17.6	22.8	62.1	74.4	80.6	31.7	66.7	62.1	56.7
PointWeb [15]	60.3	92.0	98.5	79.4	0.0	21.1	59.7	34.8	76.3	88.3	46.9	69.3	64.9	52.5
IAF-Net [11]	64.6	91.4	98.6	81.8	0.0	34.9	62.0	54.7	79.7	86.9	49.9	72.4	74.8	52.1
KPConv [10]	67.1	92.8	97.3	82.4	0.0	23.9	58.0	69.0	81.5	91.0	75.4	75.3	66.7	58.9
PointTransformer [16]	70.4	94.0	98.5	86.3	0.0	38.0	63.4	74.3	89.1	82.4	74.3	80.2	76.0	59.3
MM-3DScene(Ours)	71.9	94.6	98.6	87.1	0.0	44.2	62.9	79.2	90.7	81.7	74.3	81.4	79.3	60.3

Table d. Semantic segmentation results on S3DIS dataset evaluated on Area 5.

tic segmentation. In this section, we utilize other backbone networks for verifying the generalization ability of our MM-3DScene.

H3DNet object detection. We apply our MM-3DScene pretrained framework on H3DNet [14] which is a more powerful network using hybrid geometric primitives based on VoteNet [6]. As shown in Table. b, MM-3DScene improves the H3DNet with the mAP@0.25 by 2.0 and mAP@0.5 by 1.5, exceeding the performance with VoteNet as the backbone.

Stratified Transformer semantic segmentation. We also evaluate the performance of Stratified Transformer [3] as the backbone on S3DIS semantic segmentation. We reproduce the backbone performance using its official code and report the results in Table. c. Our MM-3DScene surpasses Stratified Transformer by 1.3% mIoU. However, it comes with a high computational cost (2.5 times of MM3D-Scene + PT) and a long training time.

Discussions. Although both H3DNet [14] and Stratified Transformer [3] inherit VoteNet [6] and Point Transformer [16], and achieve decent performance, they introduce highly-engineered architectures tailored to their network-specific operations, making it difficult to evaluate the improvement made by the self-supervised frameworks. Thus, we advocate simple and classical baselines, with the goal of minimizing the influence of network architectures to better measure the performance gain *purely* from the self-supervised pretraining framework – MM-3DScene.

Moreover, both Point Transformer [16] and VoteNet [6] stand out with conspicuously excellent **efficiency**, as reflected in *model size*, *training time*, and *inference time* of Table b and Table c, which is highly important for the deployment on real applications.

Appendix D: More fine-grained quantitative results

To provide a more comprehensive analysis, we present the segmentation results of each category in Table d. We observe that most categories have different degrees of improvement over the Point Transformer [16] backbone that we use. For instance, we achieve 6.2% gain on column, 4.9% on door, 3.3% on board, and slight decrease on window and chair.

References

[1] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *arXiv preprint arXiv:1803.10091*, 2018. 3

[2] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Netw.*, 1991. 1

[3] 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 *CVPR*, 2022. 2, 3

[4] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, 2018. 3

[5] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *NeurIPS*, 2018. 3

[6] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, 2019. 2, 3

[7] 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. 3

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

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

[10] 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. 3

[11] Mingye Xu, Zhipeng Zhou, Junhao Zhang, and Yu Qiao. Investigate indistinguishable points in semantic segmentation of 3d point cloud. In *AAAI*, 2021. 3

[12] Xiaoqing Ye, Jiamao Li, Hexiao Huang, Liang Du, and Xiaolin Zhang. 3d recurrent neural networks with context fusion for point cloud semantic segmentation. In *ECCV*, 2018. 3

[13] Chris Zhang, Wenjie Luo, and Raquel Urtasun. Efficient convolutions for real-time semantic segmentation of 3d point clouds. In *3DV*, 2018. 3

[14] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primitives. In *ECCV*, 2020. 2, 3

[15] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. PointWeb: Enhancing local neighborhood features for point cloud processing. In *CVPR*, 2019. 3

[16] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021. 2, 3