Supplementary Material
Harnessing Text-to-Image Diffusion Models for Point Cloud Self-Supervised Learning

Yiyang Chen[1] Shanshan Zhao[2†] Lunhao Duan[2] Changxing Ding[1,3†] Dacheng Tao[4]
[1]South China University of Technology
[2]Alibaba International Digital Commerce Group [3]Pazhou Lab [4]Nanyang Technological University
{eeyiyangchen, sshan.zhao00, dacheng.tao}@gmail.com, lhduan@whu.edu.cn, chxding@scut.edu.cn

In this Supplementary Material, we provide additional experiments, visualization, and illustration to better evaluate our method.

S1. Additional Experiments

Projector for Feature Alignment. We conduct experiments to investigate how the number of transformer blocks in the projector affects performance. As shown in Tab. S1, when the number of blocks is set to 3, our method achieves the best performance, suggesting that increasing the number of blocks in the projector improves the effect of feature alignment. However, excessive complexity in the projector can negatively affect performance.

Table S1. Ablation study on the number of blocks in the projector.

Number of Blocks	ScanObjectNN
1	88.34
2	88.75
3	90.08
4	89.03

Only Fine-tuning Prediction Head. To make a fair comparison with the previous methods, we fine-tune both the backbone and the prediction head in our main experiments. Additionally, freezing the entire 3D backbone and updating only the prediction head can help improve the fine-tuning efficiency, so we conduct experiments and report them in Tab. S2.

As shown in Tab. S2, under this configuration, Point-BERT and our method achieve 81.64% and 85.46% accuracy, respectively, each requiring one hour of fine-tuning. This approach saves an hour compared to fine-tuning the entire backbone, although it also leads to some performance degradation.

Table S2. Ablation study on fine-tuning strategies. Time (h) refers to fine-tuning time on the PB-T50-RS setting using a single NVIDIA RTX 4090 GPU.

Method	Fine-tuning Whole Backbone		Fine-tuning Prediction Head	
	Time	Accuracy	Time	Accuracy
Point-BERT [7]	2.0	83.07	1.0	81.64
Ours	2.0	90.08	1.0	85.46

Deploying a Pre-trained Encoder in the First Stage. To verify whether deploying a pre-trained model in the first stage effectively improves performance, we conduct experiments by loading the Point-BERT weights.

From the results in Tab. S3, loading the Point-BERT weights in the first stage yields only slight improvement, as this stage primarily serves to construct the point-to-image framework rather than optimize 3D representations.

Table S3. Ablation study on deploying a pre-trained encoder in the first stage.

Method	ScanObjectNN		
	OBJ-BG	OBJ-ONLY	PB-T50-RS
Point-BERT [7]	87.43	88.12	83.07
Ours	95.18	93.63	90.08
Ours+Point-BERT	95.35	93.63	90.15

Evaluation on Outdoor Datasets. We conduct additional experiments on SemanticKITTI [1] with a SparseConvNet [2] backbone to evaluate our method on outdoor LiDAR datasets.

As shown in Tab. S4, our method improves the baseline performance from 68.6% to 69.5% mIoU, demonstrating its effectiveness in real-world outdoor scenarios.

Integration with 3D Intra-modal Self-supervised Loss. Our method adopts only a cross-modal alignment loss, while methods like I2P-MAE [9] and ReCon [3] incorporate both intra-modal and inter-modal self-supervision. This difference in supervision may explain the performance vari-

[†] Correspondence author.

Table S4. Semantic segmentation results on the SemanticKITTI dataset measured by mIOU (%).

Method	SemanticKITTI
SparseConvNet [2]	68.6
Ours	**69.5**

Table S5. Classification accuracy (%) on the three subsets of the ScanObjectNN dataset and ModelNet40 dataset.

Method	ScanObjectNN			ModelNet40
	OBJ-BG	OBJ-ONLY	PB-T50-RS	
Point-BERT [7]	87.43	88.12	83.07	92.7
Point-FEMAE [8]	95.18	93.29	90.22	**94.0**
ReCon [3]	95.18	93.63	90.63	**94.0**
Ours	95.18	93.63	90.08	93.7
Ours+Point-BERT	**95.53**	**93.98**	**90.67**	93.9

ation observed on some datasets.

To test this hypothesis, we integrate the Point-BERT 3D intra-modal loss into our method in the second stage. As shown in Tab. S5, this single-modal self-supervision helps PointSD further improve its performance.

S2. Visualization and Illustration

Visualization of Point-to-image Generation. As shown in Fig. S1, we visualize the point cloud and the corresponding rendered image in the left two columns, respectively, and the results generated with different seeds are shown in the five right columns. From top to bottom, the point clouds in the third and sixth rows are generated by mixing the point clouds from rows 1 and 2, and rows 4 and 5, respectively. The results demonstrate that our point-to-image framework can generate the corresponding images with point clouds as the condition, enabling us to extract SD features containing semantics.

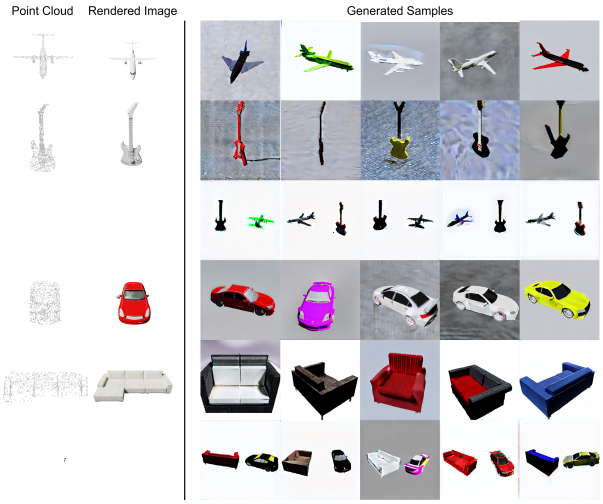

Figure S1. Visualization of point-to-image generation results.

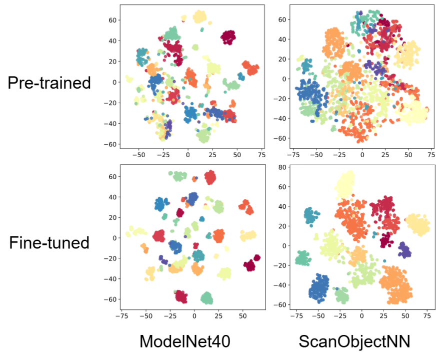

Figure S2. t-SNE visualization on ModelNet40 and ScanObjectNN PB-T50-RS datasets.

t-SNE Visualization of Features from 3D Backbone. In Fig. S2, we achieve t-SNE [5] visualization on the feature distribution extracted by our pre-trained and fine-tuned models on ModelNet40 [6] and ScanObjectNN PB-T50-RS [4] datasets. The results show that 1) Our pre-trained models can extract discriminative features on the ModelNet40 dataset without fine-tuning. 2) Our fine-tuned models can yield more discriminative features on both datasets. 3) ScanObjectNN PB-T50-RS is a real-world dataset containing background noise, while our model is pre-trained on synthetic data, making it harder for the model to separate different classes of samples in feature space without fine-tuning.

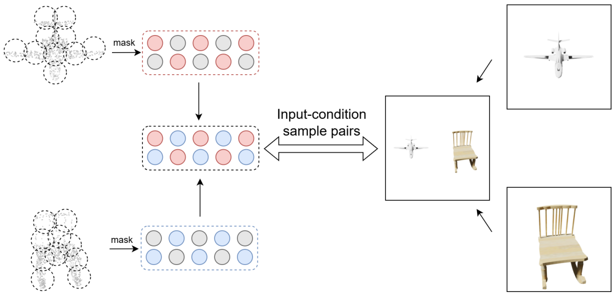

Figure S3. Illustration of augmented training samples construction.

Illustration of the Augmentation Strategy. We show our augmentation strategy in Fig. S3. Given two point cloud samples, we first divide them into a series of patches respectively and then mask out part of them to mix. For the corresponding two image samples, we stitch them along the width directly. This augmentation strategy aids in learning more robust 3D representations.

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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307, 2019. 1

[2] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018. 1, 2

[3] Zekun Qi, Runpei Dong, Guofan Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In *International Conference on Machine Learning (ICML)*, 2023. 1, 2

[4] 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. 2

[5] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 2

[6] 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

[7] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19313–19322, 2022. 1, 2

[8] Yaohua Zha, Huizhen Ji, Jinmin Li, Rongsheng Li, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Towards compact 3d representations via point feature enhancement masked autoencoders. *arXiv preprint arXiv:2312.10726*, 2023. 2

[9] Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21769–21780, 2023. 1