Appendix

For a thorough understanding of our Point Prompt Training (PPT), we have compiled a detailed Appendix. The table of contents below offers a quick overview and will guide to specific sections of interest.

Contents

A. Related Work

3D scene understanding. Deep learning techniques for understanding 3D scenes using neural networks can be broadly classified into three categories based on their approach to handling point clouds: projection-based, voxel-based, and point-based methods. Projection-based approaches involve projecting 3D points onto multiple image planes and utilizing 2D CNN-based backbones for feature extraction [12, 53, 56, 84]. In contrast, voxel-based methods convert point clouds into regular voxel representations to facilitate 3D convolutions [63, 83]. The efficiency of these methods is further enhanced through the use of sparse convolution techniques [16, 28]. Unlike the previous two, point-based methods operate directly on point clouds [71, 72, 90, 121] and have recently begun incorporating transformer-based architectures [30, 107, 122]. Following previous pre-training literatures [35, 108, 110], we train on the voxel-based SparseUNet [16], which is more efficient and allows large-scale training.

3D representation learning. Deep neural networks are notoriously data-hungry, and scaling up the pre-training data has become a promising path to learning robust and transferrable representations. Unlike in 2D vision, where large-scale curated datasets are readily available [3, 23], data collection and annotation in 3D vision is much more costly, and the scale of point cloud datasets are quite limited [2, 21]. Regarding 3D representation learning, previous works commonly pre-train on a single dataset [32, 35, 77, 78, 103, 110], which limits the potential to benefit from the scaling law [46]. As the first attempt towards scaling up the pre-training data, a recent work [108] first explored unsupervised pre-training on merged data (ScanNet [21] and ArkitScenes [5]). However, as the distributions of 3D datasets vary much, naively merging them could be sub-optimal, which is studied in this work.

Towards large-scale pre-training. In order to scale up pre-training and learn better representations, two popular topics in 2D vision is to exploit uncurated data in the wild [27, 85, 91, 92], and to better utilize the data in hand [60, 104, 109, 111, 123]. Yet the former is not applicable to 3D data, and the latter has been well-studied in previous works [35, 108, 110]. The topic of joint learning across multiple datasets has also been explored in some works related to 2D scene understanding [47, 95, 99, 117, 127] and 3D object detection [119], but while they focus on direct evaluation on the target dataset (similar to domain generalization [11, 14, 75, 97]). Our work targets more on generalized representation learning in both supervised and unsupervised settings. Moreover, the high variation between 3D datasets, and the sparse and heavily long-tailed nature, also add to the difficulty of 3D joint training.

Prompt learning. In an effort to improve the generalizability of pre-trained models on downstream tasks, prompting was originally proposed in natural language processing [59]. The prompt templates could be heuristic designed [7, 31, 79], automatically generated [25, 81], or learned as task-specific parameters [19, 29, 38, 55, 61]. We rephrase the latter one as prompt learning. In 2D vision, prompt learning has become a popular parameter-efficient technique to adapt pre-trained models to specific *downstream tasks* [4, 26, 42, 45, 118, 126]. Our work, instead, tackles *pre-training* directly. Prompt learning is regarded as a dataset-specific adapter to allow the model to resolve the domain shift between pre-training datasets separately, and learn the optimal overall representation.

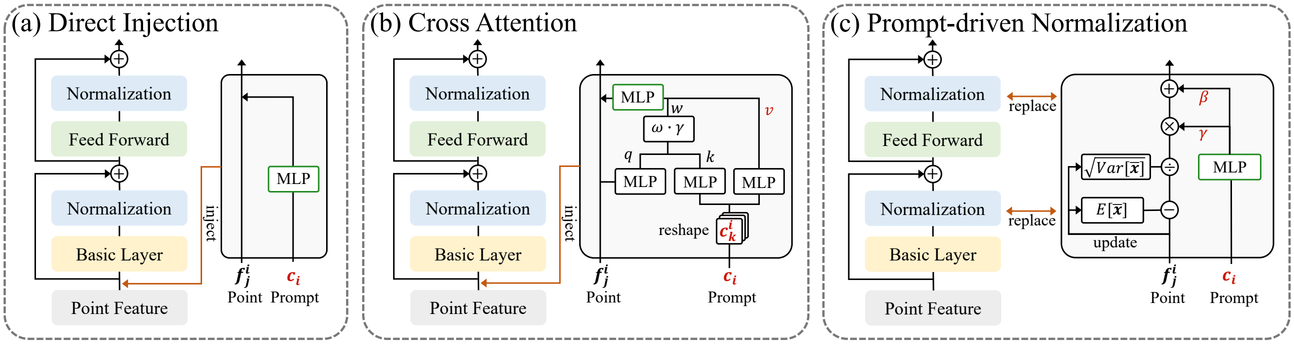


Figure 3. **Domain prompt adapters.**

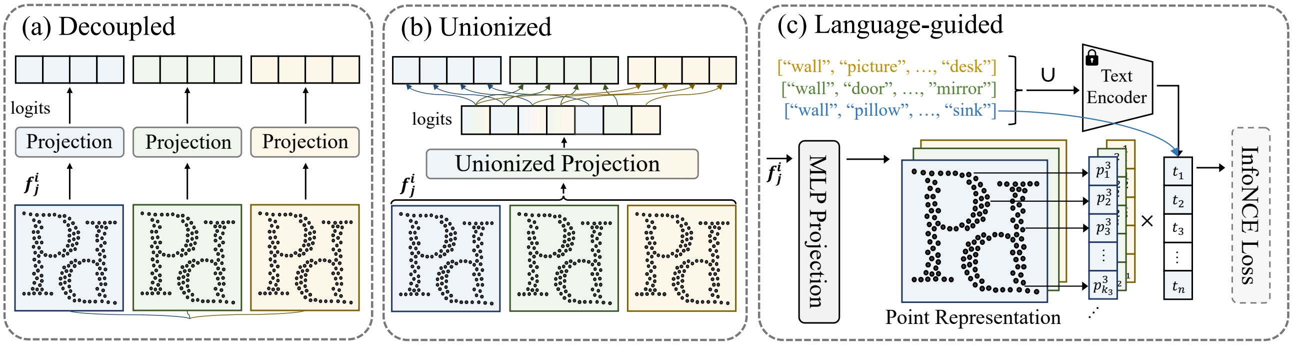


Figure 4. **Categorical alignment approaches.**

B. Alternative Designs

In this section, we provide a comprehensive overview and visual demonstration of the implementation details for proposed alternative designs for Point Prompt Training (PPT).

B.1. Domain Prompt Adapter

To address the challenges of adapting the model to different dataset domains, we introduce domain prompt adapters along with zero-initialization techniques. Fig. 3 serves as a visual guide, showcasing the implementation of each domain prompt adapter discussed in the main paper. Notably, the zero-initialized layers are highlighted with a green box.

Direct Indiction. Fig. 3a showcases the process of Direct Injection. This approach inserts a direct injection adapter at the beginning of each basic block. The domain prompt is added to the point embedding after undergoing a zero-initialized linear projection within each direct injection.

Cross Attention. As shown in Fig. 3b, the cross-attention adapter can be seen as an extension of the direct injection adapter. The domain prompts c_i splits into k independent prompt embeddings of identical shape, serving as the reference for cross-attention with each point. Attention operations [107] occur between query vectors from each point and key value vectors from the prompt embeddings. The output, post-projection by a zero-initialized linear layer, is added to the point embedding.

Prompt-driven Normalization. Fig. 3c illustrates the Prompt-driven Normalization (PDNorm) approach. In this case, each normalization layer is replaced with PDNorm, which enables the adaptation of the backbone to the specific domain context. PDNorm projects the domain prompt onto the scale-shift vector using a zero-initialized linear layer, and these domain-aware vectors are subsequently applied to the normalized feature embedding.

B.2. Categorical Alignment

To address the issue of inconsistency within the category space during supervised multi-dataset ergiergistic training, various categorical alignment strategies are explored in the main paper. Fig. 4 provides a detailed illustration of these categorical alignment methods.

Decoupled. Fig. 4a shows the decoupled approach for categorical alignment. In this method, a separate prediction head is employed for each dataset. After the shared backbone extracts the point embeddings, they are fed into the prediction head specific to the corresponding dataset's domain. Loss calculation is performed within the category space corresponding to each domain.

Unionzied. Fig. 4b presents the unified method of categorical alignment. Unlike the decoupled strategy, point embeddings are not split based on their respective domains. Instead, they pass through a unified prediction head that projects the point representations into the unified category

Datasets	#C	wall	floor	cabinet	bed	chair	sofa	table	door	window	bookshelf	bookcase	picture	counter	desk	shelves	curtain	dresser	pillow	mirror	ceiling	refrigerator	television	shower curtain	nightstand	toilet	sink	lamp	bathtub	garbagebin	board	beam	column	clutter	otherstructure	otherfurniture	otherprop
ScanNet	20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓					✓		✓		✓	✓		✓							✓	
S3DIS	13	✓	✓			✓	✓	✓	✓	✓		✓									✓										✓	✓	✓	✓			
Struct.3D	25	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓							✓	✓	✓

Table 7. **Categorical settings.**

Pre-training (joint)		**Fine-tuning** (ScanNet)		**Fine-tuning** (S3DIS)	
Config	**Value**	**Config**	**Value**	**Config**	**Value**
optimizer	SGD	optimizer	SGD	optimizer	SGD
scheduler	cosine decay	scheduler	cosine decay	scheduler	poly
learning rate	0.5	learning rate	0.5	learning rate	0.1
weight decay	1e-4	weight decay	1e-4	weight decay	1e-4
momentum	0.8	momentum	0.9	momentum	0.9
batch size	24	batch size	12	batch size	12
datasets	ScanNet (2)	datasets	ScanNet	datasets	S3DIS
	S3DIS (1)		-		-
	Struct.3D (4)		-		-
warmup iters	6k	warmup epochs	40	warmup epochs	0
iters	120k	epochs	800	epochs	3000

Table 8. **Training settings.**

space. The logit value of each category is predicted within this space. However, we still restrict each point prediction space to its corresponding domain's category space during loss calculation.

Language-guided. Fig. 4c demonstrates the language-guided approach. Here, we leverage a CLIP [74] pre-trained text encoder to extract the text embedding of each category. The alignment process involves aligning each point representation with the text embedding of its category. This alignment is facilitated by utilizing InfoNCE [65] loss as the alignment criterion. Specifically, we calculate the similarity between the point representation and the text embedding. The resulting similarity matrix is multiplied by a logit scaler (100) [74] to determine the logit value of each category, and cross-entropy loss is computed accordingly.

C. Additional Experiments

C.1. Experimental Settings

Data. We conduct PPT joint (pre)-training on three datasets: ScanNet v2 [21], S3DIS [2], and Structured3D [124]. The ScanNet v2 dataset consists of 1,613 scene scans reconstructed from RGB-D frames. It is partitioned into 1,201 scenes for training, 312 scenes for validation, and 100 scenes for benchmark testing. Point clouds in this dataset are sampled from the vertices of reconstructed meshes, and each sampled point is assigned a semantic label from a set of 20 categories. The S3DIS dataset comprises 271 rooms from six areas in three distinct buildings. Model performance evaluation is typically done using results from Area 5 and 6-fold cross-validation (result available in Tab. 11). Unlike ScanNet v2, points in the S3DIS dataset are densely sampled on the surfaces of the meshes and annotated into 13 categories. Structured3D is a synthetic photo-realistic dataset containing 3.5K house designs created by professional designers. It is annotated with the same set of 40 categories as the NYU Depth V2 [82] dataset. The dataset is divided into 3,000 scenes for training, 250 scenes for validation, and 250 scenes for testing. We further split the 3,500 scenes into approximately 20,000 rooms and project the panoramic image of each room into a 3D point cloud for training. Following the approach in Swin3D [116], the frequency of occurrence of the 40 categories is counted. Categories with frequencies less than 0.001 are filtered out, and end up with a reduced set of 25 categories for perception. Similar to Swin3D, we include the categories table of the three datasets in Tab. 7 to provide a clear reference to the category relation across the three datasets.

Training. The default joint (pre-)training and fine-tuning setting is in Tab. 8. During joint training, we follow a sampling strategy where the batched point cloud for each iteration was sampled from a single dataset. The sampling ratio is determined based on the best performance necessary iteration number for each dataset. This approach ensures that each dataset contributes to the training process in proportion to its optimal performance. Consequently, the total number of training iterations is equal to the sum of the best performance necessary iteration numbers for all the datasets involved as mentioned above. Furthermore, we observe that

Config	Value	
name	SpUNet-S (default)	SpUNet-L
patch embed depth	1	1
patch embed channels	32	96
patch embed kernel size	5	5
encode depths	[2, 3, 4, 6]	[6, 6, 12, 6]
encode channels	[32, 64, 128, 256]	[96, 192, 384, 768]
encode kernel size	3	3
decode depths	[2, 2, 2, 2]	[2, 2, 2, 2]
decode channels	[256, 128, 64, 64]	[768, 384, 192, 192]
decode kernel size	3	3
pooling stride	[2, 2, 2, 2]	[2, 2, 2, 2]
params	39M	412M

Table 9. **Backbone settings.**

using a larger batch size leads to more stable performance during training. Our fine-tuning follows the practice of supervised SparseUNet training setting from *Pointcept* [17].

Backbone. We validate the effectiveness of our Point Prompt Training by leveraging SparseUNet [16], optimized by *Pointcept* [17] with the *SpConv* [18] library. The utilization of SparseUNet was chosen due to its notable advantages in terms of speed and memory efficiency. The specific configuration of the backbone is outlined in Tab. 9, with our primary results based on the widely employed SpUNet-S, featuring 39 million parameters. Additionally, we explore the impact of employing a larger-scale backbone with 412 million parameters, denoted as SpUNet-L. The analysis of PPT's properties with the larger-scale backbone is discussed in Sec. C.4.

C.2. Additional Pilot Study

Naive joint-training with varied sampling ratios. In the pilot study, which is conducted in the main paper, we perform training experiments by naively pairwise merging ScanNet, S3DIS, and Structure3D datasets, as well as training on a combination of all datasets. Subsequently, we evaluate the model's performance on each individual dataset. The determination of the sampling ratio is based on the necessary iteration number for achieving the best performance on each dataset. Consequently, we select a sampling ratio of 4:2:1 for Structure3D, ScanNet, and S3DIS accordingly.

Concerns naturally arise regarding the potential impact of a larger sampling rate for the Structured3D point cloud. It is possible that this could lead to the model bias toward the more frequently witnessed domain, exacerbating performance degradation in other datasets rather than improving naively joint training. To investigate this further, we conduct an additional pilot study, exploring different sampling rates during naively joint training.

Tab. 10 provides an illustration of two representative sampling ratios: 4:2:1 and 1:1:1. The experimental results indicate that although increasing the sampling rate of ScanNet and S3DIS data with the balanced sampling ratio 1:1:1 slightly alleviated the performance degradation, the negative transfer effect remained significant in our vanilla setting. These findings further underscore the challenges associated with achieving effective collaborative learning across multiple datasets in the 3D domain.

C.3. Additional Results

S3DIS 6-fold semantic segmentation. Tab. 11 presents the results of our 6-fold cross-validation semantic segmentation experiment on the S3DIS dataset. For each fold, we withhold one area of S3DIS and perform PPT joint training using the remaining data along with the ScanNet and Structured3D datasets. We then evaluate and report the model's performance on the withheld area data. The average of these results represents the 6-fold cross-validation results. Notably, Point Prompt Training achieves a significant improvement in SparseUNet performance on this benchmark, with a notable 12.7% increase, establishing a new SOTA result.

Error bar-supplemented results. As a supplement to the main paper, we present the full semantic segmentation results in the main paper in Tab. 12, in which we supplement the error bar derived from five independent runs. The mean-std result of the ScanNet test mIoU is not available since multiple submissions are not allowed.

C.4. Additional Ablation Study

Backbone up-scaling. Tab. 13a presents our investigation into the impact of scaling up the backbone using multi-dataset Point Prompt Training (PPT). As a baseline, we evaluate the performance of SpUNet-S and SpUNet-L trained solely on the ScanNet dataset. Our observations indicate that, in this setup, increasing the model capacity results in significant overfitting. However, when PPT is introduced with a larger-scale data source, the issue of overfitting is mitigated, and a larger-scale backbone yields improved model performance.

To provide a visual representation of these findings, Fig. 5 illustrates the loss curves for the training and validation splits of the four experiments. The entire training period was evenly divided into 100 epochs, and the average loss on the training and validation splits was calculated at the end of each epoch to generate the curves. It is noteworthy that SpUNet-L with PPT exhibits a more favorable loss curve compared to SpUNet-S with PPT, while the opposite trend is observed in the absence of PPT.

However, it is important to consider that expanding the depth and dimension of convolution-based models results in a significant increase in parameters. As a result, transformer-based methods are better suited for exploring model capacity expansion. Nevertheless, it is worth noting that transformer-based methods currently have limitations in terms of speed and memory consumption. As part of future work, optimizing the efficiency of transformer-

data	ScanNet	S3DIS	Struct.3D	all
ScanNet	<u>72.2</u>	69.5	67.2	69.7
S3DIS	64.7	<u>65.4</u>	63.6	63.5
Struct.3D	73.9	73.7	<u>74.5</u>	72.4

(a) Sampling Ratio 1:1:1

data	ScanNet	S3DIS	Struct.3D	all
ScanNet	<u>72.2</u>	71.8	65.9	68.9
S3DIS	64.1	<u>65.4</u>	62.8	63.3
Struct.3D	73.7	74.2	<u>74.5</u>	72.9

(b) Sampling Ratio 4:2:1

Table 10. **Naive joint-training with varied sampling ratios.**

split	Area1	Area2	Area3	Area4	Area5	Area6	PPT	Scratch
mIoU	83.01	65.39	87.09	74.13	72.73	86.42	**78.13**	<u>65.4</u>
mAcc	90.25	75.58	91.83	84.01	78.22	92.47	**85.39**	-
allAcc	93.48	88.34	94.56	90.84	91.45	94.45	**92.19**	-

Table 11. **S3DIS semantic segmentation 6-fold cross-validation results.**

		ScanNet [21]		ScanNet200 [76]		S3DIS Area5 [2]	
Methods	Params.	Val mIoU	Test mIoU	Val mIoU	Test mIoU	mIoU	mAcc
SparseUNet [16]	39.2M	<u>72.2</u>	<u>73.6</u>	<u>25.0</u>	<u>25.3</u>	<u>65.4</u>	<u>71.7</u>
+ PPT Sup. (joint)	41.0M	75.4 ±0.46	-	-	-	71.9 ±0.32	77.5 ±0.38
+ PPT Sup. (f.t.)	41.0M	76.2 ±0.18	-	31.7 ±0.22	-	72.4 ±0.21	77.9 ±0.30

Table 12. **Error bar-supplemented results.**

backbone PPT	S -	L -	S ✓	L ✓
results	73.4	72.9	75.7	**75.8**

(a) Backbone Up-scaling

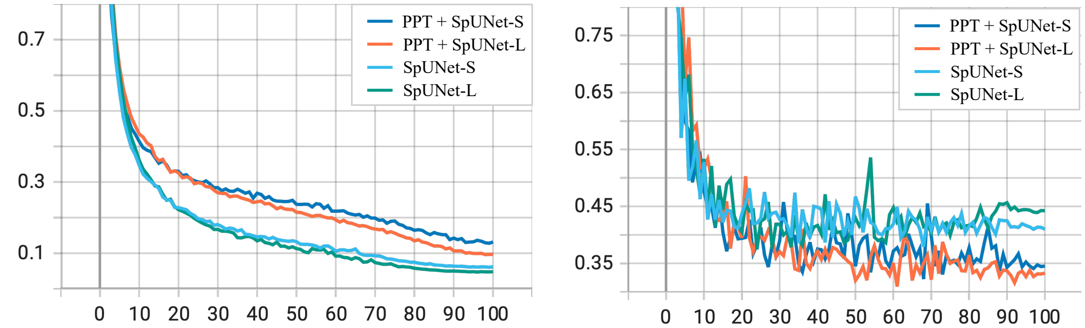
backbone shared	S -	S ✓
results	75.3	**75.7**

(b) Shared Domain Prompt

backbone head	S Linear	S LCA
results	73.4	**74.2**

(c) LCA as Prediction Head

Table 13. **Additional ablation.**



(a) Training Loss Curve

(b) Validation Loss Curve

Figure 5. **Loss curve.**

based backbones after scaling up remains a topic worth investigating.

Shared domain prompt. In Tab. 13b, validating the effectiveness of globally shared domain prompts in comparison to independent ones across different backbone blocks. Similar to the conclusion in VPT [42], we observe that employing block-wise independent domain prompts resulted in a decline in performance. We attribute this to the complexity introduced by having separate domain prompts for each block, leading to overfitting. This aligns with the observations from our ablation study in the main paper, where scaling up prompt dimensions had a similar degradation.

LCA as prediction head. We introduce Language-guided Categorical Alignment (LCA) as a method to align the category spaces across multiple datasets with a unified category-text embedding. This alignment strategy can also be employed as a segmentation prediction head within a standard single dataset training process. By considering the scaled similarity between point embedding and category-text embedding as the predicted logit value, LCA serves as an effective prediction head. In Tab. 13c, we compare the performance of the standard linear prediction head with LCA as the prediction head. The experimental results demonstrate that LCA can also enhance model performance

Methods	Year	Val	Test
○ PointNet++ [72]	2017	53.5	55.7
○ 3DMV [20]	2018	-	48.4
○ PointCNN [57]	2018	-	45.8
○ SparseConvNet [28]	2018	69.3	72.5
○ PanopticFusion [64]	2019	-	52.9
○ PointConv [105]	2019	61.0	66.6
○ JointPointBased [15]	2019	69.2	63.4
○ KPConv [90]	2019	69.2	68.6
○ PointASNL [113]	2020	63.5	66.6
○ SegGCN [54]	2020	-	58.9
○ RandLA-Net [37]	2020	-	64.5
○ JSENet [39]	2020	-	69.9
○ FusionNet [120]	2020	-	68.8
○ PTv1 [122]	2021	70.6	-
○ FastPointTransformer [67]	2022	72.4	-
○ SratifiedTranformer [50]	2022	74.3	73.7
○ PointNeXt [73]	2022	71.5	71.2
○ PTv2 [107]	2022	75.4	74.2
○ LargeKernel3D [13]	2023	73.5	73.9
○ PointMetaBase [58]	2023	72.8	71.4
○ PointConvFormer [106]	2023	74.5	74.9
○ OctFormer [100]	2023	75.7	76.6
○ Swin3D [116]	2023	75.5	-
● + Supervised [116]	2023	76.7	77.9
○ MinkUNet [16]	2019	72.2	73.6
● + PC [110]	2020	74.1	-
● + CSC [35]	2021	73.8	-
● + MSC [108]	2024	75.5	-
● + GC [96]	2024	75.7	-
● + PPT (Ours)	2024	76.4	76.6
○ OA-CNNs [69]	2024	76.1	75.6
○ PTv3 [17]	2024	77.5	77.9
● + PPT (Ours)	2024	**78.6**	**79.4**

Table 14. **ScanNet V2 semantic segmentation.**

Methods	Year	Area5	6-fold
○ PointNet [71]	2017	41.1	47.6
○ SegCloud [89]	2017	48.9	-
○ TanConv [88]	2018	52.6	-
○ PointCNN [57]	2018	57.3	65.4
○ ParamConv [101]	2018	58.3	-
○ PointWeb [121]	2019	60.3	66.7
○ HPEIN [43]	2019	61.9	-
○ KPConv [90]	2019	67.1	70.6
○ GACNet [98]	2019	62.9	-
○ PAT [115]	2019	60.1	-
○ SPGraph [52]	2018	58.0	62.1
○ SegGCN [54]	2020	63.6	-
○ PAConv [112]	2021	66.6	-
○ PTv1 [122]	2021	70.4	65.4
○ StratifiedTransformer [50]	2022	72.0	-
○ PointNeXt [73]	2022	70.5	74.9
○ PTv2 [107]	2022	71.6	73.5
○ PointMetaBase [58]	2023	72.0	77.0
○ Swin3D [116]	2023	72.5	76.9
● + Supervised [116]	2023	74.5	79.8
○ MinkUNet [16]	2019	65.4	65.4
● + PC [110]	2020	70.3	-
● + CSC [35]	2021	72.2	-
● + MSC [108]	2023	70.1	-
● + GC [96]	2024	72.0	-
● + PPT (Ours)	2024	72.7	78.1
○ PTv3 [17]	2024	73.4	77.7
● + PPT (Ours)	2024	**74.7**	**80.8**

Table 15. **S3DIS semantic segmentation.**

in the context of standard single dataset segmentation tasks.

D. Additional Comparision

In this section, we expand upon the combined results table for semantic segmentation (Tab. 3 and Tab. 4) from our main paper, offering a more detailed breakdown of results alongside the respective publication years of previous works. This comprehensive result table is designed to assist readers in tracking the progression of research efforts in 3D representation learning. Marker ○ refers to the result from a model trained from scratch, and ● refers to the result from a pre-trained model.

D.1. Indoor Semantic Segmentation

We conduct a detailed comparison of pre-training technologies and backbones on the ScanNet v2 [21] (see Tab. 14) and S3DIS [2] (see Tab. 15) datasets. ScanNet v2 comprises 1,513 room scans reconstructed from RGB-D frames, divided into 1,201 training scenes and 312 for validation. In this dataset, model input point clouds are sampled from the vertices of reconstructed meshes, with each point assigned a semantic label from 20 categories (e.g., wall, floor, table). The S3DIS dataset for semantic scene parsing includes 271 rooms across six areas from three buildings. Following a common practice [72, 89, 122], we withhold area 5 for testing and perform a 6-fold cross-validation. Different from ScanNet v2, S3DIS densely sampled points on mesh surfaces, annotated into 13 categories. Consistent with standard practice [72]. We employ the mean class-wise intersection over union (mIoU) as the primary evaluation metric for indoor semantic segmentation.

D.2. Outdoor Semantic Segmentation

We extend our comprehensive evaluation of pre-training technologies and backbones to outdoor semantic segmentation tasks, focusing on the SemanticKITTI [6](see Tab. 16) and NuScenes [8] (see Tab. 17) datasets. SemanticKITTI is derived from the KITTI Vision Benchmark Suite and consists of 22 sequences, with 19 for training and the remaining 3 for testing. It features richly annotated LiDAR scans, offering a diverse array of driving scenarios. Each point in this dataset is labeled with one of 28 semantic classes, encompassing various elements of urban driving environments. NuScenes, on the other hand, provides a large-scale dataset for autonomous driving, comprising 1,000 diverse urban driving scenes from Boston and Singapore. For outdoor semantic segmentation, we also employ the mean

Methods	Year	Val	Test
○ SPVNAS [87]	2020	64.7	66.4
○ Cylinder3D [128]	2021	64.3	67.8
○ PVKD [36]	2022	-	71.2
○ 2DPASS [114]	2022	69.3	72.9
○ PTv2 [107]	2022	70.3	72.6
○ WaffleIron [70]	2023	68.0	70.8
○ SphereFormer [51]	2023	67.8	74.8
○ RangeFormer [49]	2023	67.6	73.3
○ MinkUNet [16]	2019	63.8	-
● + PPT (Ours)	2024	71.4	-
○ OA-CNNs [69]	2024	70.6	-
○ PTv3 [17]	2024	70.8	74.2
● + M3Net [62]	2024	72.0	75.1
● + PPT (Ours)	2024	**72.3**	**75.5**

Table 16. **SemanticKITTI semantic segmentation.**

Methods	Year	Val	Test
○ SPVNAS [87]	2020	77.4	-
○ Cylinder3D [128]	2021	76.1	77.2
○ PVKD [36]	2022	-	76.0
○ 2DPASS [114]	2022	-	80.8
○ PTv2 [107]	2022	80.2	82.6
○ SphereFormer [51]	2023	78.4	81.9
○ RangeFormer [49]	2023	78.1	80.1
○ MinkUNet [16]	2019	73.3	-
● + PPT (Ours)	2024	78.6	-
○ OA-CNNs [69]	2024	78.9	-
○ PTv3 [17]	2024	80.4	82.7
● + PPT (Ours)	2024	**81.2**	**83.0**

Table 17. **NuScenes semantic segmentation.**

class-wise intersection over union (mIoU) as the primary evaluation metric for outdoor semantic segmentation.