# Towards Foundation Models for 3D Scene Understanding: Instance-Aware Self-Supervised Learning for Point Clouds

Supplementary Material

## A. Additional Implementation details

**Details of SCR** Spatial Clustering Regularization (SCR) plays a key role in *PointINS* by enforcing local geometric coherence among points, enabling the model to learn instance-level geometric reasoning. We provide the full procedure of SCR in Algorithm 1. The process consists of two stages: (i) global feature-based grouping and (ii) local spatial refinement. First, we apply K-means clustering to partition points into $K$ coarse semantic groups. This step leverages the strong semantic awareness preserved in self-supervised backbones. Next, for each segment $S_k$, we compute ODR-regularized predicted centroids and construct a $k$-nearest-neighbor graph using two constraints: (1) neighbor count $k_{\text{nn}}$ and (2) maximum neighbor distance $\tau_d$. This pruning ensures that connections only form between spatially consistent points rather than long-range neighbors. We then apply a standard BFS algorithm to decompose the graph into multiple connected components. Components smaller than a minimum size threshold $\tau_{\text{min}}$ are discarded to avoid noisy groupings. Each remaining component is treated as a pseudo-instance, from which a refined centroid is computed. The target offsets are then defined as the displacement from each point to its assigned pseudo-centroid. These targets supervise the student model via self-distillation, promoting stable local centroid alignment rather than random spatial drift.

**Pre-taining Settings** We summarize all hyperparameter configurations in Tab. 1. For single-dataset pre-training and downstream fine-tuning, we use a single NVIDIA H200 GPU. For multi-dataset pre-training, we use two H200 GPUs in Distributed Data Parallel (DDP) mode. Following Sonata [11] and DOS [1], we concatenate multi-scale features from the last three encoder stages rather than using only the final stage.

**Multi-Dataset Pre-training** To enhance cross-domain generalization, we pre-train *PointINS* jointly on multiple datasets. For outdoor settings, we follow standard practice and train on the combined corpus of nuScenes [3], SemanticKITTI [2], and Waymo [9], totaling approximately 116k point clouds. A single unified model is trained across all datasets using the same architecture and hyperparameters as in single-dataset pre-training. For indoor settings, we pre-train on Structured3D [13], ScanNet [5], and S3DIS [8], comprising roughly 24k point clouds. Following prior works [1, 11], we scale up model capacity to better handle the increased structural diversity of indoor environments, expanding the encoder to $[3, 3, 3, 12, 3]$ blocks with channel widths $[48, 96, 192, 384, 512]$. This configuration yields a 108M-parameter model, compared to 38M in the default setup.

| Config | Outdoor | Indoor |
|---|---|---|
| optimizer | AdamW | AdamW |
| scheduler | Cosine | Cosine |
| learning rate | 2e-3 | 4e-3 |
| weight decay | 4e-2 | 4e-2 |
| batch size | 16 | 16 |
| Datasets | nuScenes / Sem.Kitti | Scannet / Scannet200 / S3DIS |
| Mask Ratio | 0.7 | 0.7 |
| Mask Size | 1 m | 40 cm |
| warmup ratio | 0.05 | 0.05 |
| training epochs | 50 | 800/800/3000 |
| $\alpha_{zipf}$ | 1.3 | 0.1 / 1.3 / 0.1 |
| warmup ratio of $L_{\text{off}}$ | 0.1 | 0.1 |
| $\lambda_{\text{off}}$ | 0.25 | 0.25 |
| $K$ (K-means) | 20 | 20 |
| $iter$ (K-means) | 10 | 10 |
| $k_{nn}$ | 20 | 150 |
| $\tau_d$ | 1 m | 120 cm |
| $\tau_{min}$ | 10 | 30 |
| Distribution ($\mathcal{D}$) | Uniform(0, 1) | Uniform(0, 1) |
| Distribution ($\mathcal{M}$) | LogNormal($\mu$=0, $\sigma$=0.76) | Gamma($a$=0.24, $\theta$=2.53) |

Table 1. Pretraining settings for indoor and outdoor point clouds.

**Indoor Instance Segmentation** We evaluate *PointINS* on three indoor benchmarks. **ScanNet** [5] contains 1,613 RGB-D scans with 3D instance annotations, split into 1,201 training, 312 validation, and 100 test scenes. **ScanNet200** [10] extends ScanNet with fine-grained labels over 200 semantic categories. Following standard protocol, we report instance segmentation using the 18 canonical instance classes shared with ScanNet. **S3DIS** [8] consists of 271 indoor scenes across six areas annotated with 13 semantic classes, all of which are evaluated for instance segmentation. We adopt the common *Area-5* protocol, where Area 5 serves as the test split and the remaining areas for training. We report mean Average Precision (mAP) as the primary metric. $\text{AP}_{25}$ and $\text{AP}_{50}$ denote AP at 25% and 50% IoU thresholds, while AP averages scores from 50% to 95% IoU (step size 5%).

**Outdoor Panoptic Segmentation** We evaluate on two large-scale LiDAR benchmarks. **SemanticKITTI** [2] contains 22 driving sequences captured by a 64-beam LiDAR sensor with 19 semantic classes; sequences 00–10 (excluding 08) are used for training, 08 for validation, and 11–21 for testing. **nuScenes** [3] includes 1000 urban driving scenes from Boston and Singapore collected with a 32-beam LiDAR sensor. Following [7], we evaluate 16 merged semantic classes. Panoptic segmentation performance is

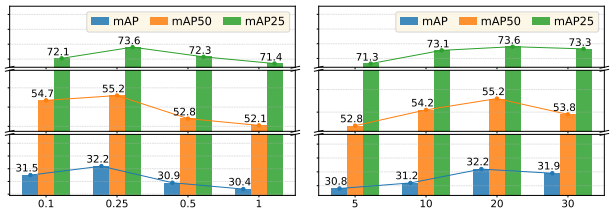
measured using the Panoptic Quality (PQ):

$$\text{PQ} = \underbrace{\frac{\sum_{(i,j)\in \text{TP}} \text{IoU}(i,j)}{|\text{TP}|}}_{\text{Segmentation Quality (SQ)}} \times \underbrace{\frac{|\text{TP}|}{|\text{TP}| + \frac{1}{2}(|\text{FP}| + |\text{FN}|)}}_{\text{Recognition Quality (RQ)}}, \quad (1)$$

where SQ measures segmentation accuracy and RQ measures instance recognition. We additionally report SQ and RQ separately.

## B. Additional Experiments

**Effect of $\lambda_{\text{off}}$ and $K$** We study the impact of two key parameters in our method: the offset loss weight $\lambda_{\text{off}}$ and the number of K-means clusters $K$ used in SCR. The results are shown in Fig. 1. Regarding the loss weight, $\lambda_{\text{off}} = 0.25$ achieves the optimal performance among all tested values. For K-means clustering, using $K = 20$ yields the best performance, which intuitively aligns with the semantic category distribution of the dataset. More broadly, $K$ controls a fundamental trade-off: too small value risks merging semantically distinct but feature-similar regions, while too large value tends to over-segment individual objects into spurious parts. Empirically, we found moderate values of $K$ is sufficient, as most scenes contain a limited number of distinct semantic concepts.




(a) Loss weight $\lambda_{\text{off}}$ (b) Number of K-Means clusters $K$

Figure 1. Results of different parameter settings on ScanNet.

**Effect of Warmup Ratio of $L_{\text{off}}$** In Tab. 2, we study how different warmup ratios for the offset loss affect instance segmentation performance on ScanNet. Compared to no warmup, introducing a short warmup phase significantly improves results, with the best performance observed at a ratio of 0.1. This suggests that the model benefits from first establishing stable semantic representations before learning offsets. However, increasing the warmup duration beyond 0.1 leads to a gradual performance drop, as excessive delay reduces the effective training time for geometric reasoning. Overall, these results highlight the importance of gradually introducing the offset loss to stabilize early optimization while still allowing sufficient training for instance awareness.

**Runtime Analysis** Since *PointINS* builds upon DOS [1], the additional offset branch and two regularization steps introduce extra computation during pre-training. Overall, the total pre-training time increases by approximately 25% (e.g. from 20 to 25 hours on ScanNet [5] and from 24 to 29 hours on nuScenes [3]), measured on a single GPU.

| Warmup Ratio | mAP | $AP_{50}$ | $AP_{25}$ |
|---|---|---|---|
| 0.0 | 30.8 | 53.7 | 72.8 |
| 0.1 | 32.1 | 55.2 | 73.6 |
| 0.2 | 31.7 | 54.6 | 73.1 |
| 0.4 | 31.4 | 53.5 | 72.4 |
| 0.6 | 30.1 | 52.6 | 71.6 |
| 0.8 | 28.8 | 49.9 | 68.1 |

Table 2. Results of different warmup ratio of $L_{\text{off}}$ on ScanNet

**Object Detection** To further verify the generalization of our approach, we evaluate *PointINS* on the nuScenes object detection benchmark [3] using CenterPoint [12] as the detector, with results reported in Tab. 3. Under decoder probing, the pre-trained encoder is frozen and only the remaining detector components are trained. Under finetuning, all model weights are optimized end-to-end. In both settings, *PointINS* outperforms existing SSL approaches by a significant margin, demonstrating its transferability across diverse downstream tasks and representing a promising step toward holistic 3D foundational perception.

| Method | OD Prob. | | OD 1% | |
|---|---|---|---|---|
| | mAP | NDS | mAP | NDS |
| Sonata [11] | 44.6 | 55.0 | 41.3 | 52.8 |
| DOS [1] | 55.4 | 61.5 | 49.0 | 58.0 |
| *PointINS* | **56.7** | **62.5** | **50.8** | **60.2** |

Table 3. OD Prob.: Object detection under decoder probing protocol. OD 1%: Object detection finetuning on 1% annotations.

**Cross-dataset Probing** Beyond single- and multi-dataset pre-training, we further evaluate *PointINS* under a cross-dataset probing setting, where the model is pre-trained on one dataset and linearly probed on another. As shown in Tab. 4, *PointINS* consistently outperforms existing SSL baselines across both transfer directions, confirming that the instance-aware representations learned by *PointINS* generalize robustly across different outdoor scene layouts.

| Method | SK→Nu | | | Wa→Nu | | |
|---|---|---|---|---|---|---|
| | PQ | SQ | RQ | PQ | SQ | RQ |
| Sonata [11] | 31.0 | 72.5 | 40.5 | 36.9 | 75.4 | 47.1 |
| DOS [1] | 46.4 | 78.9 | 57.5 | 51.4 | 80.3 | 62.6 |
| *PointINS* | **56.7** | **80.1** | **59.4** | **54.8** | **81.9** | **66.1** |

Table 4. Results on panoptic segmentation under cross-dataset probing setting. SK: SemanticKITTI [2], Nu: nuScenes [3], Wa: Waymo Open Dataset [9]

## C. Unsupervised Instance Segmentation

We further assess whether *PointINS* produces useful instance-aware representations without any downstream supervision. Instead of training an instance segmentation

**Algorithm 1** Spatial Clustering Regularization (SCR)

**Require:** Teacher features $\mathbf{F} = \{f_i\}_{i=1}^{N}$, coordinates $\{x_i\}_{i=1}^{N}$, ODR-regularized offsets $\{\tilde{\mathcal{O}}_i\}_{i=1}^{N}$

**Require:** Hyperparameters: number of neighbors $k_{nn}$, distance threshold $\tau_d$, minimum instance size $\tau_{min}$

**Ensure:** Pseudo-instance targets $\{\mathcal{O}_i^*\}_{i=1}^{N}$

1: **Predict centroids:**

$$\hat{c}_i \leftarrow x_i + \tilde{\mathcal{O}}_i$$

2: **Global feature grouping:**

$$\{S_1, \ldots, S_K\} \leftarrow \mathrm{KMeans}(\mathbf{F}; K, iter)$$

3: **Local spatial clustering per segment:**
For each segment $S_k$:
(a) Compute $k$ nearest neighbors for each point $i \in S_k$ using Euclidean distance in centroid space:

$$\mathcal{N}(i) = \mathrm{kNN}(\hat{c}_i, k_{nn})$$

Retain edges only if distance is below threshold:

$$j \in \mathcal{N}(i) \quad \textbf{iff} \quad \|\hat{c}_i - \hat{c}_j\|_2 < \tau_d$$

(b) Extract connected components using Breadth-First Search (BFS):

$$\mathcal{I}_k = \mathrm{BFS}(\mathcal{N})$$

(c) Remove small components (noise filtering):

$$\mathcal{I}_k \leftarrow \{I_{k,j} \in \mathcal{I}_k \mid |I_{k,j}| \geq \tau_{min}\}$$

4: **Compute refined centroids:**

$$\bar{c}_{k,j} = \frac{1}{|I_{k,j}|} \sum_{i \in I_{k,j}} \hat{c}_i$$

5: **Generate new offset targets:**

$$\mathcal{O}_i^* = \bar{c}_{k,j} - x_i, \quad \forall i \in I_{k,j}$$

6: **return** $\{\mathcal{O}_i^*\}_{i=1}^{N}$

head, we directly use the offsets predicted by the pretrained model and perform BFS-based clustering on the offset-shifted centroids to generate instance proposals. The resulting clusters are matched to ground-truth instances via Hungarian assignment for evaluation. As shown in Tab. 5, *PointINS* substantially outperforms classical unsupervised baselines such as HDBSCAN [4] and Felzenszwalb clustering [6], demonstrating strong geometric reasoning and robust instance separation ability even without task-specific optimization. These results highlight the broader utility of our approach beyond standard self-supervised settings. In addition to quantitative results, we present qualitative evaluations of unsupervised instance segmentation in Fig. 2.

| Method | mAP | $AP_{50}$ | $AP_{25}$ |
|---|---|---|---|
| HDBSCAN [4] | 1.7 | 4.2 | 16.4 |
| Felzenszwalb [6] | 1.2 | 2.3 | 13.4 |
| *PointINS* | 10.8 | 18.7 | 43.4 |

Table 5. Results of unsupervised instance segmentation on ScanNet.

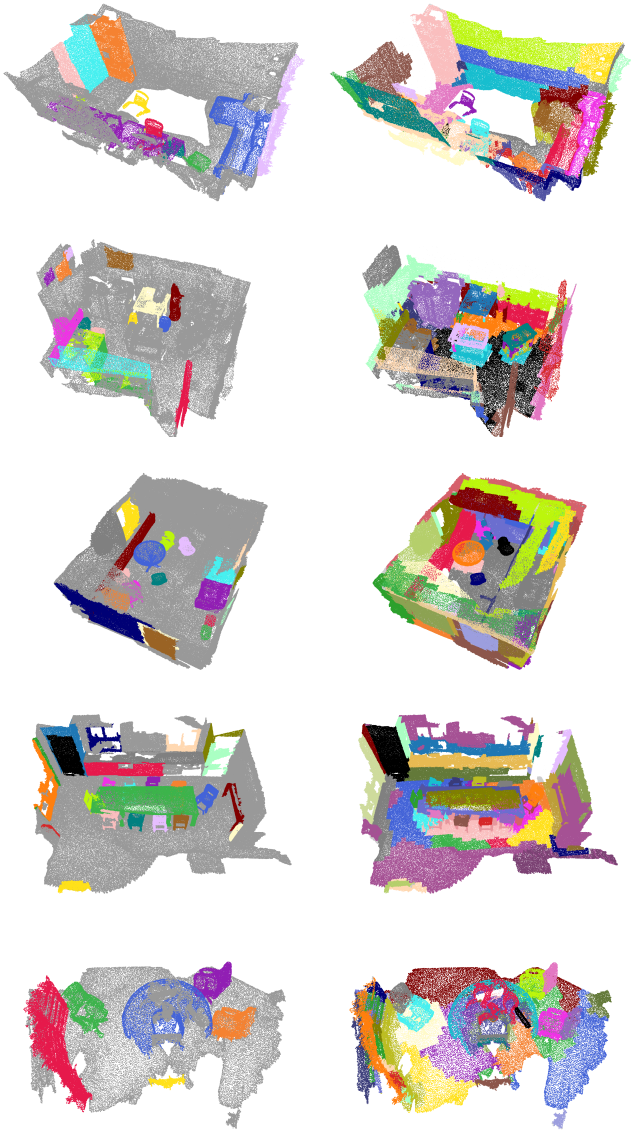


Figure 2. Qualitative results of unsupervised instance segmentation. Left: ground-truth instance labels. Right: predictions from *PointINS* obtained directly from offset clustering without any downstream supervision.

## References


[1] Mohamed Abdelsamad, Michael Ulrich, Bin Yang, Miao Zhang, Yakov Miron, and Abhinav Valada. Dos: Distilling observable softmaps of zipfian prototypes for self-supervised point cloud representation learning. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2026. 1, 2

[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 *International Conference on Computer Vision (ICCV)*, pages 9297–9307, 2019. 1, 2

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11621–11631, 2020. 1, 2

[4] Ricardo J.G.B. Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2013. 3

[5] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2

[6] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59:167–181, 2004. 3

[7] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *RAL*, 7(2):3795–3802, 2022. 1

[8] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4558–4567, 2018. 1

[9] Jieru Mei, Alex Zihao Zhu, Xinchen Yan, Hang Yan, Siyuan Qiao, Liang-Chieh Chen, and Henrik Kretzschmar. Waymo open dataset: Panoramic video panoptic segmentation. In *European Conference on Computer Vision (ECCV)*, pages 53–72. Springer, 2022. 1, 2

[10] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *European Conference on Computer Vision (ECCV)*, 2022. 1

[11] Xiaoyang Wu, Daniel DeTone, Duncan Frost, Tianwei Shen, Chris Xie, Nan Yang, Jakob Engel, Richard Newcombe, Hengshuang Zhao, and Julian Straub. Sonata: Self-supervised learning of reliable point representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1, 2

[12] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (CVPR)*, 2021. 2

[13] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *European Conference on Computer Vision (ECCV)*, 2020. 1