Fine-grained Image-to-LiDAR Contrastive Distillation with Visual Foundation Models

Yifan Zhang and Junhui Hou*
Department of Computer Science, City University of Hong Kong
yzhang3362-c@my.cityu.edu.hk;jh.hou@cityu.edu.hk

Abstract

Contrastive image-to-LiDAR knowledge transfer, commonly used for learning 3D representations with synchronized images and point clouds, often faces a self-conflict dilemma. This issue arises as contrastive losses unintentionally dissociate features of unmatched points and pixels that share semantic labels, compromising the integrity of learned representations. To overcome this, we harness Visual Foundation Models (VFMs), which have revolutionized the acquisition of pixel-level semantics, to enhance 3D representation learning. Specifically, we utilize off-the-shelf VFMs to generate semantic labels for weakly-supervised pixel-to-point contrastive distillation. Additionally, we employ von Mises-Fisher distributions to structure the feature space, ensuring semantic embeddings within the same class remain consistent across varying inputs. Furthermore, we adapt sampling probabilities of points to address imbalances in spatial distribution and category frequency, promoting comprehensive and balanced learning. Extensive experiments demonstrate that our approach mitigates the challenges posed by traditional methods and consistently surpasses existing image-to-LiDAR contrastive distillation methods in downstream tasks. Code is available at https://github.com/Eaphan/OLIVINE.

1 Introduction

LiDAR sensors deliver critical information in the 3D environment, crucial for applications such as autonomous driving [78, 75, 79]. State-of-the-art neural networks have shown promising performance on point clouds processing, which rely on extensive annotated datasets [29, 77, 17]. However, the process of annotating point clouds is not only time-consuming but also costly, presenting significant challenges in terms of scalability and practicality [36]. Self-supervision offers a solution by leveraging vast quantities of unlabeled data to pre-train networks and subsequently fine-tuning them with a smaller, labeled dataset. This approach significantly reduces the reliance on extensive annotated data sets [10].

A prevalent method for learning 3D representations involves contrastive pixel-to-point knowledge transfer, using synchronized and calibrated images and point clouds. PPKT [39] enables a 3D network to derive extensive knowledge from a pre-trained 2D image backbone through a pixel-to-point contrastive loss. This entire pre-training process necessitates no annotations for either images or point clouds. Then SLidR [50] employs superpixels to cluster pixels and points from visually coherent regions, leading to a more meaningful contrastive task. Building on this, Seal [38] utilizes semantically rich superpixels generated by visual foundation models and introduces temporal consistency regularization across point segments at different times. Meanwhile, HVDistill [73] innovates by implementing cross-modality contrastive distillation that integrates both image-plane and bird-eye views.

*Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-4078

Unfortunately, existing contrastive distillation methods are hindered by several critical limitations. **Firstly**, a "self-conflict" issue arises during pre-training, where (super)pixels that belong to the same category as an anchor (super)point but do not directly correspond are simply treated as negative samples (see Fig.1(a)). This approach neglects the inherent semantic connections within the same category, leading to conflicts in the learning process where beneficial relationships might be disregarded. This problem is magnified by the contrastive loss's intrinsic hardness-aware characteristic, which results in the most substantial gradient influences derived from negative samples that are semantically the most similar [59, 41]. While ST-SLidR [41] has introduced a semantically tolerant loss to mitigate this issue, the absence of a robust high-level semantic understanding could not fundamentally change the intrinsic hardness-aware nature of the contrastive loss. **Secondly**, conventional sampling methods for point-pixel pairs fail to consider significant category imbalances or variations in point densities relative to their distance from sensors [39]. For example, bicycles comprise only 1.47% of annotations in the nuScenes-lidarseg dataset, whereas drivable surfaces make up 37.66%. This oversight can result in a skewed representation of the environment, where dominant categories or densely populated areas are over-represented, impacting the model's effectiveness and fairness.

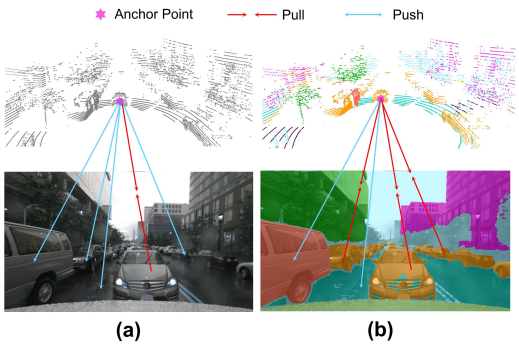

Figure 1: Illustration of (**a**) self-conflict that exists in conventional pixel-to-point contrastive distillation and (**b**) our weakly supervised contrastive distillation.

In this study, we address the "self-conflict" issue by leveraging supervised contrastive distillation enhanced with weak semantic labels generated by VFMs (Visual Foundation Models). VFMs like SAM (Segment Anything Model), trained on expansive datasets, has revolutionized computer vision by simplifying the acquisition of pixel-level semantics. These models are exceptionally adaptable, enabling the direct derivation of semantic labels through specified prompts without the need for retraining. As depicted in Fig.1(b), with these weak labels, we draw the embeddings of an anchor point and corresponding pixels of the same class closer together, while distancing the anchor from "negative" pixels of differing classes. Furthermore, given that representation of samples in the same class can vary significantly across different batches, we introduce semantic-guided consistency regularization to enhance 3D representation learning. This approach structures the feature space by modeling each class with a von Mises-Fisher distribution and making point features adhere closely to their respective distributions.

Considering the challenges posed by category imbalance and the non-uniform distribution of point clouds, we propose a density and category-aware sampling strategy. This method accounts for both the density of points and the frequency of their categories. By adjusting the sampling probabilities of different anchor points, we enhance the quality of learned 3D representations, particularly for points that fall into minority categories or are situated in areas of low density.

Extensive experiments reveal that our pre-training approach outperforms state-of-the-art 3D self-supervised methods on both the nuScenes and SemanticKITTI datasets. **The primary contributions** of this work are summarized as follows: **1**) To tackle the challenge of "self-conflict", we utilize off-the-shelf VFMs to generate semantic labels for weakly-supervised pixel-to-point contrastive distillation. **2**) We introduce semantic-guided consistency regularization to cultivate a meaningful and structured feature space. **3**) We develop an innovative sampling strategy for point-pixel pairs that considers both the category frequency and spatial density of points.

2 Related Work

3D Representation Learning. Recent advancements in 3D self-supervised learning have closely paralleled innovations in the image domain, extending these methods to diverse 3D contexts such as object-level point clouds [49, 60, 23], indoor scenes [61, 12, 81, 68, 9, 32, 69, 58], and outdoor settings [34, 4, 71, 42, 18]. These techniques are grounded in contrastive learning [66, 71, 42], mask modeling [72], and other pretext tasks [4, 76]. PPKT [39] utilized the InfoNCE loss to facilitate the 3D network in distilling rich knowledge from the 2D image backbone. Sautier et al.[50] pioneered a superpixelto-superpoint contrastive loss for self-supervised 2D-to-3D representation distillation.

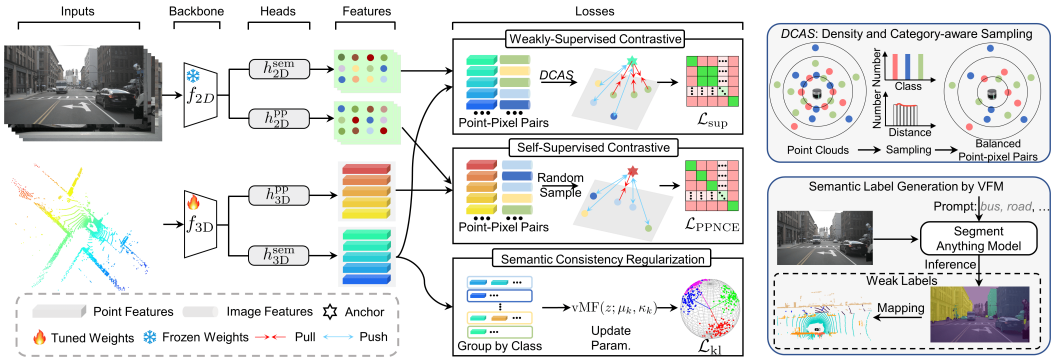

Figure 2: The overall pipeline of our proposed OLIVINE. The pipeline starts with feature extraction via a trainable 3D backbone and a pre-trained 2D backbone, followed by feature alignment in a common space. The learning is driven by weakly-supervised contrastive distillation with coarse semantic labels, self-supervised distillation of randomly sampled point-pixel pairs, and semantic consistency regularization through the von Mises-Fisher distribution. Besides, our approach is also characterized by the novel sampling strategy of point-pixel pairs addressing spatial and category distribution imbalances.

Building on this, Mahmoud et al. [41] enhanced the approach by incorporating a semantically tolerant contrastive constraint and a class-balancing loss. Liu et al.[38] further refined these techniques through semantic-aware spatial and temporal consistency regularization, advancing feature learning. Moreover, Zhang et al. [73] explored cross-modality contrastive distillation across not only the image plane but also the bird-eye views.

Visual Foundation Models. The advent of powerful visual neural networks, trained on extensive datasets [46, 27] or through cutting-edge self-supervised learning techniques [7, 11, 21], catalyzed significant advancements within the community. Notably, the Segment Anything Model (SAM) [27] initiated a new paradigm in general-purpose image segmentation, demonstrating remarkable zero-shot transfer capabilities across a variety of downstream tasks. Building on this, Grounded-SAM [47] enhanced the model by incorporating elements from Grounding-DINO [37], an open-set object detector capable of recognizing and classifying previously unseen objects during training [37]. In our work, we leverage the inherent semantic awareness of these VFMs [84] to generate weak semantic labels, which are crucial for our supervised contrastive distillation framework.

3D Scene Understanding. Traditional approaches to 3D scene understanding primarily utilize paradigms based on raw points [55, 16, 8, 62], voxels [17, 13], range views [29, 56], and multi-view fusion [19, 67]. Despite the effectiveness of these methods in capturing detailed environmental features, they are significantly constrained by their dependence on large-scale annotated data. Acquiring and annotating this data is both time-consuming and costly, limiting the scalability of 3D segmentation models [36]. To reduce the reliance on large annotated datasets, recent studies have also turned to semi-supervised [31, 22], weakly-supervised [35, 15], and active learning techniques [40, 65].

3 Proposed Method

Overview. As depicted in Fig. 2, our method, namely OLIVINE, integrates a visual foundation model for pre-training with paired point clouds and images. Feature extraction is conducted using a trainable 3D backbone for point clouds and a pre-trained 2D backbone for images, with these features then mapped into a common feature space via decoupled projection heads for both point-pixel level and category-level contrastive distillation. The representation learning in OLIVINE is driven by three objectives: weakly-supervised contrastive distillation using coarse semantic labels to identify positive pairs by category, self-supervised contrastive distillation applied to randomly sampled point-pixel pairs, and a regularization framework based on the von Mises-Fisher distribution to ensure semantic consistency. Additionally, we address imbalances in spatial distribution and category frequency through a targeted sampling strategy, ensuring a balanced representation in the learning process.

Notation. Let $P = \{p_1, p_2, ..., p_N | p_i \in \mathbb{R}^3\}$ be a point cloud consisting of N points collected by a LiDAR sensor, and $\mathcal{I} = \{I_c \| \ c = 1, ..., N_{\text{cam}}\}$ multi-view images captured by N_{cam} synchronized cameras, where $I_c \in \mathbb{R}^{H \times W \times 3}$ is a single image with height H and width W.

https://doi.org/10.52202/079017-4078

3.1 Baseline Architecture

We follow the existing work [39] to perform the basic point-to-pixel contrastive distillation, based on which we build our whole pipeline. Starting with point cloud and image inputs, we employ distinct encoders for feature extraction. The 3D features are extracted using an encoder $f_{3\mathrm{D}}(\cdot) : \mathbb{R}^{N\times 3} \rightarrow \mathbb{R}^{N\times C_{3\mathrm{D}}}$, which processes the point clouds to produce features of dimension $C_{3\mathrm{D}}$ per point. For image features, we use an encoder $f_{2D}(\cdot) : \mathbb{R}^{H\times W\times 3} \rightarrow \mathbb{R}^{H'\times W'\times C_{2\mathrm{D}}}$, initialized with weights from pre-trained image models. This setup facilitates knowledge transfer from the 2D domain to the 3D domain through contrastive learning. For the computation of contrastive loss, we design trainable projection heads, $h^{\mathrm{pp}}_{2\mathrm{D}}$ for 2D features and $h^{\mathrm{pp}}_{3\mathrm{D}}$ for 3D features, both aligning the features into a unified dimensional space. Specifically, the 3D projection head $h^{\mathrm{pp}}_{3\mathrm{D}}$ is a linear layer with ℓ_2-normalization, converting 3D features to a normalized C-dimensional space. Similarly, the 2D projection head $h^{\mathrm{pp}}_{2\mathrm{D}}$, a convolution layer with a 1×1 kernel followed by a bi-linear interpolation layer adjusting the spatial dimensions by a factor of 4, also applies ℓ_2-normalization.

Utilizing the calibration matrix, we establish dense point-to-pixel correspondences $\{\boldsymbol{F}^{3\mathrm{D}}_i, \boldsymbol{F}^{2\mathrm{D}}_i\}^{M}_{i=1}$, where $\boldsymbol{F}^{3\mathrm{D}}_i$ and $\boldsymbol{F}^{2\mathrm{D}}_i$ represent the paired features of points and images for the i-th pair, with M denoting the total count of such valid pairs. Previous methods achieve cross-modal knowledge transfer by attracting positive pairs and repelling negative pairs within the feature space, employing the InfoNCE loss [43]. The point-pixel level contrastive loss is defined as

$$\mathcal{L}_{\mathrm{PPNCE}} = -\frac{1}{M_s}\sum_{i=1}^{M_s}\log\left[\frac{\exp(\langle \boldsymbol{F}^{3\mathrm{D}}_i, \boldsymbol{F}^{2\mathrm{D}}_i\rangle/\tau)}{\sum_{j=1}^{M_s}\exp(\langle \boldsymbol{F}^{3\mathrm{D}}_i, \boldsymbol{F}^{2\mathrm{D}}_j\rangle/\tau)}\right], \tag{1}$$

where τ is the temperature factor, M_s is the number of sampled corresponding point-pixel pairs, $\langle\cdot,\cdot\rangle$ denotes the scalar product measure the similarity between features.

3.2 Weakly-supervised Contrastive Distillation

Existing methods [39, 50, 38] often directly treat unmatched points and pixels that share semantic labels as negative pairs. This practice overlooks the intrinsic semantic connections within the same categories, leading to potential conflicts in the learning process where beneficial relationships are ignored. To address this, we utilize the Segment Anything Model, which adeptly interprets and translates semantic cues from text prompts into precise semantic segmentation of images. This application of SAM allows us to generate high-quality semantic labels without repetitive training, enhancing the learning process. We represent these labels as $Y^{\mathrm{co}} = \{y^{\mathrm{co}}_i\}^{M}_{i=1}$, where each label corresponds to a specific point-pixel pair.

In point-pixel level contrastive loss, the pixels that belong to the same category but do not correspond to the given anchor point are taken as negative samples (see Eq. (1)). Therefore, we argue that the 2D and 3D features used for weakly-supervised contrastive learning, which take the category information into consideration, should differ from the features $\boldsymbol{F}^{3\mathrm{D}}$ and $\boldsymbol{F}^{2\mathrm{D}}$ that represent the individual points and pixels. To address this issue, we apply another two heads $h^{\mathrm{sem}}_{2\mathrm{D}} : \mathbb{R}^{H'\times W'\times C_{2\mathrm{D}}} \rightarrow \mathbb{R}^{H\times W\times C}$ and $h^{\mathrm{sem}}_{3\mathrm{D}} : \mathbb{R}^{N\times C_{3\mathrm{D}}} \rightarrow \mathbb{R}^{N\times C}$ to extract the semantic-level feature embeddings $\boldsymbol{G}^{2\mathrm{D}}$ and $\boldsymbol{G}^{3\mathrm{D}}$.

For the sampled points and pixels, we use their semantic labels to identify positive and negative pairs. Positive pairs are defined as point and pixel features that share the same semantic label, whereas negative pairs are those with different labels [26]. The weakly-supervised contrastive loss is defined as

$$\mathcal{L}_{\mathrm{sup}} = -\frac{1}{M_s}\sum_{i=1}^{M_s}\log\left[\frac{1}{|A(i)|}\sum_{a\in A(i)}\frac{\exp(\langle \boldsymbol{G}^{3\mathrm{D}}_i, \boldsymbol{G}^{2\mathrm{D}}_a\rangle/\tau)}{\sum_{j=1}^{M_s}\exp(\langle \boldsymbol{G}^{3\mathrm{D}}_i, \boldsymbol{G}^{2\mathrm{D}}_j\rangle/\tau)}\right], \tag{2}$$

where $A(i)$ denotes the set of indices of matched point-to-pixel pairs in the batch that have the same class with i-th point-pixel pair, and $|A(i)|$ indicates its cardinality.

3.3 Semantic-guided Consistency Regularization

We advocate that the construction of latent semantic structures in feature space could enhance representation learning. By leveraging semantic labels derived from SAM inference, we organize points with identical semantic labels into coherent groups. This grouping promotes feature consistency within these semantic categories, thereby stabilizing the learning of feature representations across varied data instances and yielding structured feature space.

Distribution Assumption. Intuitively, the point features extracted by the projection head $h_{\text{3D}}^{\text{sem}}$ from the same class should exhibit similarity in the feature space. For the purposes of contrastive learning, these features are normalized to exist on the unit hypersphere. Consequently, we model the point features of each class k as a von Mises-Fisher (vMF) distribution $\text{vMF}(z;\mu_k,\kappa_k)$. This distribution is a spherical adaptation of the normal distribution, suitable for data constrained to a hypersphere [33]. Here, μ_k represents the mean direction, while κ_k is the concentration parameter, indicating the degree to which category features are concentrated around μ_k. The probability density function for the vMF distribution, applicable to a random C-dimensional unit vector $\boldsymbol{z}$, is formulated as follows:

$$f_C(\boldsymbol{z};\mu_k,\kappa_k) = \mathcal{K}_C(\kappa_k)\exp(\kappa_k\mu_k^\top \boldsymbol{z}), \tag{3}$$

where $\kappa \geq 0$ and $\|\mu\|_2 = 1$. The normalization constant $\mathcal{K}_C(\kappa)$ is defined as:

$$\mathcal{K}_C(\kappa_k) = \frac{\kappa_k^{C/2-1}}{(2\pi)^{C/2}\mathcal{I}_{(C/2-1)}(\kappa_k)}, \tag{4}$$

$$\mathcal{I}_{(C/2-1)}(x) = \sum_{m=0}^{\infty}\frac{1}{m!\Gamma(C/2-1+m+1)}\left(\frac{x}{2}\right)^{2m+C/2-1}, \tag{5}$$

where $\mathcal{I}_{(C/2-1)}$ is the modified Bessel function of the first kind at order $C/2-1$. The distribution exhibits a higher concentration around the mean direction μ_k as κ_k increases, and becomes uniform on the hypersphere when $\kappa_k = 0$.

Parameter Updating. Specifically, we conduct the semantic-guided consistency regularization in a *two-stage* framework. First, we update the parameter of $\text{vMF}(z;\mu_k,\kappa_k))$ with the features $\boldsymbol{G}^{\text{3D}}$ extracted from point cloud branch. During training, we obtain the statistical value of feature embeddings via the EMA (Exponential Moving Average) algorithm, following:

$$\bar{z}_k^t = \alpha\bar{z}_k^{t-1} + (1-\alpha)\bar{z}_k'^t, \tag{6}$$

where $\bar{z}_k'^t = \frac{1}{M_k}\sum_{i=1}^{M}\mathbf{1}_{\{y_i^{\text{co}}=k\}}\boldsymbol{G}_i^{\text{3D}}$ denotes the sample mean of class k at t-th mini-batch, α is the fixed smoothness coefficient. The maximum likelihood estimates of the mean direction μ_k is simply the normalized arithmetic mean:

$$\mu_k = \bar{z}_k/\bar{R}_k, \tag{7}$$

where $\bar{R}_k = ||\bar{z}_k||$. And the concentration parameter κ_k could be obtained by the solving the equation:

$$A_C(\kappa) = \bar{R}_k, \tag{8}$$

where $A_C(\kappa) = \mathcal{I}_{C/2}(\kappa)/\mathcal{I}_{C/2-1}(\kappa)$. Sra [52] proposed a simple method to estimate the κ_k:

$$\hat{\kappa}_k = \frac{\bar{R}_k(C-\bar{R}_k^2)}{1-\bar{R}_k^2}. \tag{9}$$

And we model the features of each observed point as a Spherical Dirac Delta distribution during training:

$$\delta(z-\boldsymbol{G}_i^{\text{3D}}) = \begin{cases} 0, & z \neq \boldsymbol{G}_i^{\text{3D}} \\ \infty, & z = \boldsymbol{G}_i^{\text{3D}} \end{cases} \tag{10}$$

Loss Function of Regularization. At the second step, we could perform the semantic-guided consistency regularization by pulling the points features and distribution of its corresponding category $\text{vMF}(z;\mu_k,\kappa_k)$ with Kullback-Leibler (KL) Divergence loss:

$$\begin{aligned}\mathcal{L}_{\text{kl}} &= \frac{1}{M}\sum_{i=1}^{M}D_{\text{KL}}(\delta(z-\boldsymbol{G}_i^{\text{3D}})||\text{vMF}(z;\mu_k,\kappa_k)) = \frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{K}\mathbf{1}_{\{y_i^{\text{co}}=k\}} - \log f_C(\boldsymbol{G}_i^{\text{3D}};\mu_k,\kappa_k) \\ &= \frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{K}\mathbf{1}_{\{y_i^{\text{co}}=k\}} - \log\mathcal{K}_C(\kappa_k) - \kappa_k\mu_k^\top\boldsymbol{G}_i^{\text{3D}}.\end{aligned} \tag{11}$$

In summary, the overall loss function for pre-training is written as $\mathcal{L} = \lambda_1\mathcal{L}_{\text{PPNCE}} + \lambda_2\mathcal{L}_{\text{sup}} + \lambda_3\mathcal{L}_{\text{kl}}$, where λ_1, λ_2, and λ_3 are the weights to balance the three terms.

https://doi.org/10.52202/079017-4078

3.4 Density and Category-aware Sampling Strategy

Previous methods [39] neglect the spatial distribution and category frequency imbalances in the sampling of point-pixel pairs for contrastive distillation. To overcome these challenges, we introduce a novel sampling strategy that utilizes both the distance of each point from the LiDAR sensor and the occurrence frequency of its category. First, we calculate the distance of each point in the point cloud from the LiDAR sensor. We then apply kernel density estimation (KDE) to these distances to determine the probability density function of the spatial distribution of points. Given a point, its density can be calculated using the formula based on its distance d from the LiDAR sensor:

$$f_{\mathrm{h}}(d) = \frac{1}{Mh} \sum_{i=1}^{M} K_h \left(\frac{d - d_i}{h} \right), \tag{12}$$

where d_i denotes the distance of point p_i from the LiDAR sensor, h is the bandwidth, K_h is the kernel function. This density estimation helps us understand how densely points are distributed with respect to their distance from the sensor, which is crucial for addressing areas of high point concentration that may bias the learning process.

Simultaneously, we assess the frequency of each category in the valid point-pixel pairs. By counting the occurrences of each category, we can identify which categories are underrepresented or overrepresented in the dataset.

Combining these two dimensions of analysis, we define the sampling probability for each point as inversely proportional to both its KDE-derived density and its category occurrence frequency. Mathematically, the sampling probability for a point p_i is given by:

$$\rho(p_i) = 1/(f_{\mathrm{h}}(d_i) \times |A(i)|). \tag{13}$$

By implementing this sampling strategy, we aim to ensure a more uniform representation of both spatial and categorical dimensions in our contrastive learning setup. This method reduces the risk of overfitting to dense clusters of points or to frequently occurring categories, thereby fostering a more robust and generalizable representation learning.

4 Experiments

4.1 Transfer on Semantic Segmentation

Evaluation Protocol. We evaluate the learned representations for semantic segmentation on nuScenes-lidarseg and SemanticKITTI datasets. The nuScenes-lidarseg and SemanticKITTI datasets contain 16 and 19 semantic categories for validation, respectively. We modify the network by adding a 3D convolutional layer to the pre-trained backbone as the segmentation head. Basically, we finetune the whole network on different portions of annotated data. Following previous works [50, 41], we finetune the network for 100 epochs with a batch size of 10 and 16 on SemanticKITTI and nuScenes-lidarseg, respectively. The initial learning rate of the backbone and the segmentation head are set to 0.05 and 2.0, respectively. When utilizing 1% of the annotated data, the network is fine-tuned for 100 epochs, whereas for other percentages, it is fine-tuned for 50 epochs. In another protocol, we evaluate the quality of learned representation by *linear probing*. Different from the setting of finetuning, we optimize only the added segmentation head and keep the weights of backbone $f_{3\mathrm{D}}$ frozen on the

Table 1: Comparison of various pre-training techniques for semantic segmentation tasks using either finetuning or linear probing (LP). This evaluation uses different proportions of accessible annotations from the nuScenes or SemanticKITTI datasets and presents the mean Intersection over Union (mIoU) scores on the validation set.

Initialization	Present at	nuScenes						KITTI
		LP	1%	5%	10%	25%	100%	1%
Random	-	8.1	30.30	47.84	56.15	65.48	74.66	39.50
PointContrast [66]	ECCV'20	21.90	32.50	-	-	-	-	41.10
DepthContrast [80]	ICCV'21	22.10	31.70	-	-	-	-	41.50
PPKT [39]	Arxiv'21	35.90	37.80	53.74	60.25	67.14	74.52	44.00
SLidR [50]	CVPR'22	38.80	38.30	52.49	59.84	66.91	74.79	44.60
ST-SLidR [41]	CVPR'23	40.48	40.75	54.69	60.75	67.70	75.14	44.72
Seal [38]	NeurIPS'23	44.95	45.84	55.64	62.97	68.41	75.60	46.63
Ours	NeurIPS'24	**50.09**	**50.58**	**60.19**	**65.01**	**70.13**	**76.54**	**49.38**

https://doi.org/10.52202/079017-4078

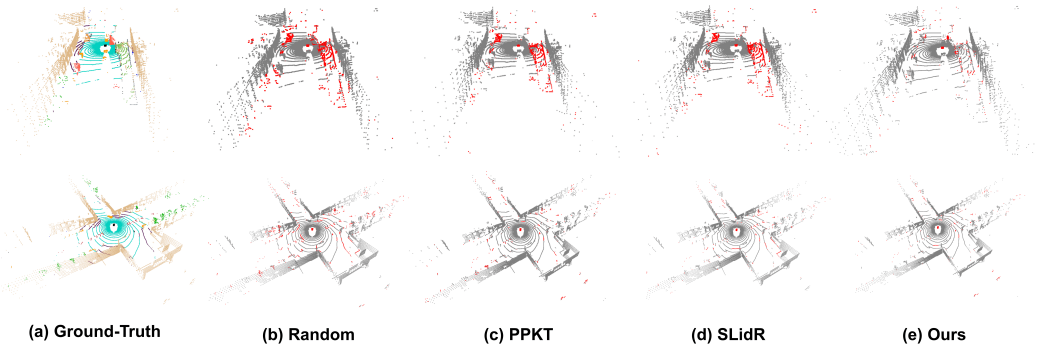

Figure 3: The visual results of various point cloud pretraining strategies, pre-trained on nuScenes and fine-tuned using merely 1% of annotated data, are displayed. To illustrate the distinctions, we mark correctly predicted areas in gray color and incorrect ones in red.

Table 2: Finetuning results on SemanticKITTI across various percentages of annotated data. The table compares the improvement achieved by our method relative to the SLidR [50].

Initialization	1%		5%		10%		20%		100%	
Random	39.5	-	45.7	-	51.5	-	56.0	-	56.9	-
SLidR [50]	44.6	+5.1	54.7	+9.0	56.3	+4.8	56.7	+0.7	57.1	+0.2
Ours	**49.4**	**+9.9**	**57.5**	**+11.8**	**60.3**	**+8.8**	**60.9**	**+4.9**	**61.1**	**+4.2**

nuScenes-lidarseg dataset. For both protocols, the training objective is a linear combination of the cross-entropy loss and the Lovász-Softmax loss [3].

Results of Linear Probing. Under the linear probing scenario, our method achieves the highest mIoU of 50.09%, surpassing the previous state-of-the-art method, Seal [38], which records a mIoU of 44.95% (see Table 1). This performance indicates a significant improvement in extracting useful features directly from pre-trained models without additional training of the 3D backbone.

Results of Fine-tuning. For the fine-tuning on nuScenes, our consistently excel, particularly at smaller data proportions. With only 1% of the training data of the nuScenes, our method achieves a mIoU of 50.58%. This trend persists across other data proportions, with our method consistently leading or closely competing with the best results, particularly at 60.19% mIoU with 5% of the data and 65.01% mIoU with 10% of the data. The qualitative results are presented in Fig. 3.

We also fine-tuned various models on the SemanticKITTI dataset to assess their performance across a spectrum of annotated data availability, from as low as 1% to full data utilization (see Table 2). Besides, as shown in Table 3, our method consistently achieves state-of-the-art performance on another six datasets.

4.2 Transfer on 3D Object Detection

Evaluation Protocol. In evaluating our pre-trained model for 3D object detection, we utilized two prominent architectures: SECOND [70] and PV-RCNN [51]. Both are built on the VoxelNet 3D backbone [82], which processes voxels via 3D sparse convolutions and includes a 2D backbone for bird's-eye-view encoding after BEV projection. The primary distinction between the architectures is in their detection heads. SECOND uses a region proposal network (RPN) directly on the 2D

Table 3: Evaluation of various pretraining methods initially trained on the nuScenes dataset and subsequently fine-tuned on multiple downstream point cloud datasets. The mIoU scores are presented as percentages (%).

Method	ScribbleKITTI		RELLIS-3D		SemanticPOSS		SemanticSTF		SynLiDAR		DAPS-3D	
	1%	10%	1%	10%	50%	100%	50%	100%	1%	10%	50%	100%
Random	23.81	47.60	38.46	53.60	46.26	54.12	48.03	48.15	19.89	44.74	74.32	79.38
PPKT [39]	36.50	51.67	49.71	54.33	50.18	56.00	50.92	54.69	37.57	46.48	78.90	84.00
SLidR [50]	39.60	50.45	49.75	54.57	51.56	55.36	52.01	54.35	42.05	47.84	81.00	85.40
Seal [38]	40.64	52.77	51.09	55.03	53.26	56.89	53.46	55.36	43.58	49.26	81.88	85.90
Ours	**44.91**	**54.96**	**53.47**	**58.21**	**55.70**	**58.51**	**56.65**	**60.42**	**46.34**	**52.78**	**83.63**	**86.84**

https://doi.org/10.52202/079017-4078

Table 4: Comparison of our method with other pre-training techniques through fine-tuning on the KITTI dataset. The results reflect the 3D object detection performance under moderate difficulty on the validation set.

Detectors	Initialization	Car	Pedestrian	Cyclist	mAP@40
SECOND [70]	Random	81.5	50.9	66.5	66.3
	PPKT [39]	81.8	51.4	68.2	67.1
	SLidR [50]	81.9	51.6	68.5	67.3
	Ours	**82.0**	**53.2**	**69.8**	**68.3**
PV-RCNN [51]	Random	84.5	57.9	71.3	71.3
	STRL [24]	84.7	57.8	71.9	71.5
	PPKT [39]	83.2	55.5	73.8	70.8
	SLidR [50]	84.4	57.3	**74.2**	71.9
	Ours	**84.8**	**59.3**	**74.2**	**72.8**

backbone, while PV-RCNN refines RPN predictions with fine-grained keypoint features, enhancing bounding box accuracy and confidence in estimations.

In the fine-tuning stage, we integrate the detection head of SECOND or PV-RCNN with the pre-trained backbone (VoxelNet). This integrated detector is then fine-tuned on the train data of KITTI [20], which includes implementations of these detectors and follows the standard training parameters specified by OpenPCDet [54]. We conduct the fine-tuning three separate times, and report the highest mean Average Precision (mAP) recorded on the KITTI validation set.

Results. The experimental results detailed in Table 4 showcase the performance of various initialization strategies. When using the SECOND architecture, our method outperforms other pre-training techniques. Starting from a baseline with random initialization, the performance improves consistently with more specialized pre-trained weights like PPKT and SLidR, and ultimately, our method achieves the highest mAP at 68.3%. Significant gains are observed across all categories, with particularly notable improvements in detecting pedestrians and cyclists. Similarly, the PV-RCNN architecture benefits from refined initialization methods. Our method again yields the highest overall mAP@40 at 72.8%, surpassing the performance of SLidR.

Remark. Compared to the semantic segmentation task, the model architecture for object detection is more complex. Besides the 3D backbone, 3D detectors typically project features to a BEV plane, followed by a 2D convolutional network and RoI operations. These crucial components were not pre-trained, which may limit the overall performance gain from our pre-training approach. It's important to note that semantic segmentation and object detection use different metrics and scales, making direct performance comparisons improper. The nature of these tasks and their evaluation criteria inherently lead to varying degrees of improvement when applying our proposed method.

Table 5: Ablation study of each component pre-trained on nuScenes and fine-tuned on nuScenes-lidarseg and SemanticKITTI. WCD: Weakly-supervised Contrastive Distillation. SCR: Sematic-guided Consistency Regularization. DCAS: Denstiy and Category-aware Sampling.

Exp.	WCD	SCR	DCAS	nuScenes						SemanticKITTI
				LP	1%	5%	10%	25%	100%	1%
1	×	×	×	36.87	37.89	53.15	60.33	67.03	74.59	44.12
2	✓	×	×	42.10	42.33	55.22	61.53	67.70	75.06	45.58
3	×	✓	×	40.58	41.99	54.49	60.98	67.69	74.88	45.67
4	✓	✓	×	44.76	44.91	56.01	**63.12**	68.74	75.48	46.60
5	✓	×	✓	44.15	43.96	55.75	62.49	68.13	75.07	46.07
6	✓	✓	✓	**47.30**	**46.12**	**57.51**	63.04	**69.39**	**76.13**	**47.35**

4.3 Ablation Studies

Effect of Key Components. In Table 5, we investigate the effect of each added component in our method. The integration of Weakly-supervised Contrastive Distillation alone yields a significant increase in performance, improving mIoU by 5.23% for linear probing. Similarly, incorporating Sematic-guided Consistency Regularization also enhances model performance, delivering a 3.71% increase in mIoU. When these components are combined, they synergistically contribute to a further mIoU gain of 7.89% for linear probing. Additionally, the application of Denstiy and Category-aware

Table 6: Comprehensive ablation studies for the key components. We report the fine-tuned results on nuScenes-lidarseg and SemanticKITTI (S.K.) datasets with 1% of the labeled data.

(a) Different types of supervision utilized for contrastive distillation.

Method	nuScenes	S.K.
Baseline	37.89	44.12
w/ Weak label	46.12	47.35
w/ GT label	57.72	51.56

(b) Different architectures of projection heads.

Heads	nuScenes	S.K.
Not Decoupled	42.59	45.46
Decoupled	46.12	47.35
Improvement	+3.53	+1.89

(c) Different distributions to model semantic features.

Distribution	nuScenes	S.K.
Deterministic	44.15	45.98
vMF	46.12	47.35
Improvement	+1.97	+1.37

(d) Different sampling strategies.

Sampling	nuScenes	S.K.
Random	44.91	46.60
Density-aware	45.33	46.77
Category-aware	45.74	46.82
DCAS	46.12	47.35

Sampling independently provides a substantial performance boost. The culmination of integrating all proposed components results in the optimal model, achieving a mIoU improvement of 10.43% for linear probing. This comprehensive analysis underscores the effectiveness of each component and their collective impact in enhancing the model's segmentation capabilities.

Potential of Supervised Contrastive Distillation. As mentioned in Section. 3, we perform weakly-supervised contrastive distillation with the pseudo-labels predicted by SAM. With the free and available models, our method learns effective 3D representations. When we replace the weak labels with the ground truth provided in nuScenes-lidarseg dataset, we can obtain a significant improvement in downstream tasks (see Table 6a). The results further demonstrate the effectiveness of supervision for cross-modal contrastive distillation and the potential of our pipeline with stronger VFMs.

Effect of the Decoupled Projection Heads. The experimental results outlined in Table 6b demonstrate the effectiveness of employing decoupled projection heads in our model. These results highlight a distinct performance enhancement when projection heads are specialized for distinct tasks — specifically, self-supervised and weakly-supervised contrastive distillation. On the nuScenes dataset, the implementation of decoupled projection heads results in a mIoU improvement of 3.53%, indicating a robust enhancement in the model's ability to generalize from the training data. Similarly, for the SemanticKITTI dataset, a gain of 1.89% in mIoU is observed, further substantiating the benefits of this architectural modification.

Effect of the vMF Distribution. The ablation study in Table 6c compares the use of deterministic (Dirac delta) and von Mises-Fisher (vMF) distributions for modeling semantic features of each class, demonstrating clear advantages with vMF on both nuScenes and SemanticKITTI datasets. Specifically, the vMF distribution, with its adjustable concentration parameter, provides a mIoU improvement of 1.97% on nuScenes and 1.37% on SemanticKITTI compared to the deterministic approach. The learned concentration parameter that represents the uncertainty helps in mitigating overfitting by providing robustness against inaccuracies in coarse semantic labels.

Effect of Different Sampling Strategies. In Table 6d, Category-aware and density-aware sampling determine the sampling probability of a point by its category frequency and distance information, respectively. These are part of a hybrid strategy we refer to as density and category-aware sampling. We found that the density and category-aware sampling strategy consistently achieves the best performance on downstream tasks.

4.4 Further Discussions

We would like to emphasize the uniqueness and advantages of our approach over existing ones:

- Previous works [39, 50, 41, 38, 73] have not solved the self-conflict problem properly. Especially, Seal [38] generates semantically coherent superpixels for distinct objects and backgrounds in the 3D scene. However, the superpoints and superpixels with the same category may still be simply considered negative pairs during contrastive learning. By

https://doi.org/10.52202/079017-4078

contrast, our method explicitly defines the points and pixels with the same semantic labels as positive pairs during weakly-supervised contrastive learning.

- Our pipeline performs knowledge distillation on two levels: self-supervised and weakly-supervised contrastive learning. To achieve this, we develop two different heads in both the image and point cloud branches to decouple the learned representation. Previous methods [39, 50, 41, 38, 73] have only attempted self-supervised contrastive distillation and have not explored using labels to guide contrastive distillation.
- The representation of samples in the same class can vary significantly across different batches during the contrastive distillation, so the model will struggle to learn stable semantic features. By making point features of the same class closely aligned, our method aims to create a more consistent and structured feature space.
- Existing methods [50, 41, 38, 73] are highly dependent on the generated superpixels. Superpixels balance asymmetries between areas with denser coverage of points and sparser areas in the contrastive loss. However, we do not need this process at all and ensure a uniform representation of both spatial and categorical dimensions by employing a novel sampling strategy.
- ST-SLidR [41] reduces the contribution of false negative samples based on superpixel-to-superpixel similarity, using 2D self-supervised features to determine semantic similarities between superpixels. By contrast, our method directly estimates the semantic labels of images with VFMs, and defines pixels and points with the same label as positive pairs.
- The purposes of using VFMs in Seal and our method are completely different. To avoid over-segmenting semantically coherent areas, Seal generates superpixels using visual foundation models (VFMs) to replace the traditional method SLIC [1]. In contrast, our method does not rely on superpixels. Although we also use VFMs, we leverage them to obtain coarse semantic labels for fine-grained contrastive distillation.

4.5 Visualization and Analysis

The T-SNE visualization shown in Fig. 4 demonstrates the efficacy of our method in achieving a more discriminative and well-separated feature distribution for each category compared to the baseline model PPKT [39]. To some extent, each category forms a distinct cluster with our method, with relatively clear boundaries separating different classes. This enhanced clustering effect underscores the benefits of our approach, which incorporates semantic supervision and applies semantic-guided consistency regularization during pre-training.

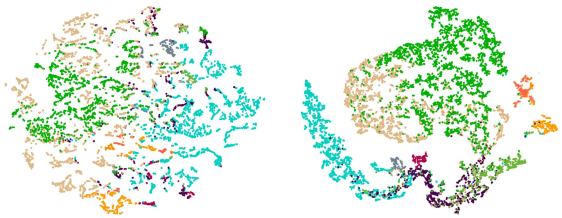

Figure 4: T-SNE visualization of the extracted point cloud features by PPKT and our OLIVINE (with head h_{3D}^{sem}). Each feature is colorized based on its ground-truth semantic labels on nuScenes dataset.

5 Conclusion

We have introduced OLIVINE, a novel method that leverages visual foundation models for fine-grained image-to-LiDAR knowledge transfer. The key ingredient of our method is utilizing the weak semantic labels generated from VFMs to avoid semantically similar negative samples and tackle the challenge of "self-conflict" challenges. We further exploit the semantic labels with semantic-guided consistency regularization, which makes embeddings of points in the same class remain consistent across varying inputs and yields structured feature space. Extensive experiments on various datasets confirm that our method achieves superior performance in downstream tasks compared to existing image-to-LiDAR contrastive distillation methods.

https://doi.org/10.52202/079017-4078

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China Excellent Young Scientists Fund 62422118, in part by the Hong Kong Research Grants Council under Grants 11219324 and 11219422, and in part by the Hong Kong Innovation and Technology Fund ITS/164/23. This work used the computational facilities provided by the Computing Services Centre at the City University of Hong Kong. Besides, we thank the anonymous reviewers for their invaluable feedback that improved our manuscript.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

[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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019.

[3] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, 2018.

[4] Alexandre Boulch, Corentin Sautier, Björn Michele, Gilles Puy, and Renaud Marlet. Also: Automotive lidar self-supervision by occupancy estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13455–13465, 2023.

[5] S.R. Bowman, L. Vilnis, O. Vinyals, A.M. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, 2016.

[6] 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*, pages 11621–11631, 2020.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[8] Chen Chen, Zhe Chen, Jing Zhang, and Dacheng Tao. Sasa: Semantics-augmented set abstraction for point-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 221–229, 2022.

[9] Runnan Chen, Youquan Liu, Lingdong Kong, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Yu Qiao, and Wenping Wang. Clip2scene: Towards label-efficient 3d scene understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7020–7030, 2023.

[10] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[11] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[12] Yujin Chen, Matthias Nießner, and Angela Dai. 4dcontrast: Contrastive learning with dynamic correspondences for 3d scene understanding. In *European Conference on Computer Vision*, pages 543–560, 2022.

[13] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21674–21683, 2023.

[14] Huixian Cheng, Xianfeng Han, and Guoqiang Xiao. Cenet: Toward concise and efficient lidar semantic segmentation for autonomous driving. In *IEEE International Conference on Multimedia and Expo*, pages 1–6, 2022.

 https://doi.org/10.52202/079017-4078

[15] Julian Chibane, Francis Engelmann, Tuan Anh Tran, and Gerard Pons-Moll. Box2mask: Weakly supervised 3d semantic instance segmentation using bounding boxes. In *European Conference on Computer Vision*, pages 681–699, 2022.

[16] Jaesung Choe, Chunghyun Park, Francois Rameau, Jaesik Park, and In So Kweon. Pointmixer: Mlp-mixer for point cloud understanding. In *European Conference on Computer Vision*, pages 620–640. Springer, 2022.

[17] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1201–1209, 2021.

[18] Chaoqun Du, Yulin Wang, Shiji Song, and Gao Huang. Probabilistic contrastive learning for long-tailed visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[19] Sudeep Fadadu, Shreyash Pandey, Darshan Hegde, Yi Shi, Fang-Chieh Chou, Nemanja Djuric, and Carlos Vallespi-Gonzalez. Multi-view fusion of sensor data for improved perception and prediction in autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2349–2357, 2022.

[20] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[22] Cheng-Ju Ho, Chen-Hsuan Tai, Yen-Yu Lin, Ming-Hsuan Yang, and Yi-Hsuan Tsai. Diffusion-ss3d: Diffusion model for semi-supervised 3d object detection. *Advances in Neural Information Processing Systems*, 36, 2024.

[23] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6535–6545, 2021.

[24] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6535–6545, 2021.

[25] Peng Jiang, Philip Osteen, Maggie Wigness, and Srikanth Saripallig. Rellis-3d dataset: Data, benchmarks and analysis. In *IEEE International Conference on Robotics and Automation*, pages 1110–1116, 2021.

[26] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

[27] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[28] Alexey Klokov, Di Un Pak, Aleksandr Khorin, Dmitry Yudin, Leon Kochiev, Vladimir Luchinskiy, and Vitaly Bezuglyj. Daps3d: Domain adaptive projective segmentation of 3d lidar point clouds. *Preprint*, 2023.

[29] Lingdong Kong, Youquan Liu, Runnan Chen, Yuexin Ma, Xinge Zhu, Yikang Li, Yuenan Hou, Yu Qiao, and Ziwei Liu. Rethinking range view representation for lidar segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 228–240, 2023.

[30] Lingdong Kong, Youquan Liu, Xin Li, Runnan Chen, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Robo3d: Towards robust and reliable 3d perception against corruptions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19994–20006, 2023.

[31] Lingdong Kong, Jiawei Ren, Liang Pan, and Ziwei Liu. Lasermix for semi-supervised lidar semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21705–21715, 2023.

[32] Lanxiao Li and Michael Heizmann. A closer look at invariances in self-supervised pre-training for 3d vision. In *European Conference on Computer Vision*, pages 656–673, 2022.

[33] Shen Li, Jianqing Xu, Xiaqing Xu, Pengcheng Shen, Shaoxin Li, and Bryan Hooi. Spherical confidence learning for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15629–15637, 2021.

[34] Guibiao Liao, Jiankun Li, and Xiaoqing Ye. Vlm2scene: Self-supervised image-text-lidar learning with foundation models for autonomous driving scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 3351–3359, 2024.

[35] Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, Abdulmotaleb El Saddik, Christian Theobalt, Eric Xing, and Shijian Lu. Weakly supervised 3d open-vocabulary segmentation. *Advances in Neural Information Processing Systems*, 36:53433–53456, 2023.

[36] Minghua Liu, Yin Zhou, Charles R Qi, Boqing Gong, Hao Su, and Dragomir Anguelov. Less: Label-efficient semantic segmentation for lidar point clouds. In *European Conference on Computer Vision*, pages 70–89, 2022.

[37] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[38] Youquan Liu, Lingdong Kong, Jun Cen, Runnan Chen, Wenwei Zhang, Liang Pan, Kai Chen, and Ziwei Liu. Segment any point cloud sequences by distilling vision foundation models. *Advances in Neural Information Processing Systems*, 36, 2024.

[39] Yueh-Cheng Liu, Yu-Kai Huang, Hung-Yueh Chiang, Hung-Ting Su, Zhe-Yu Liu, Chin-Tang Chen, Ching-Yu Tseng, and Winston H Hsu. Learning from 2d: Contrastive pixel-to-point knowledge transfer for 3d pretraining. *arXiv preprint arXiv:2104.04687*, 2021.

[40] Yadan Luo, Zhuoxiao Chen, Zijian Wang, Xin Yu, Zi Huang, and Mahsa Baktashmotlagh. Exploring active 3d object detection from a generalization perspective. In *The Eleventh International Conference on Learning Representations*, 2023.

[41] Anas Mahmoud, Jordan SK Hu, Tianshu Kuai, Ali Harakeh, Liam Paull, and Steven L Waslander. Self-supervised image-to-point distillation via semantically tolerant contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7102–7110, 2023.

[42] Lucas Nunes, Rodrigo Marcuzzi, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. Segcontrast: 3d point cloud feature representation learning through self-supervised segment discrimination. *IEEE Robotics and Automation Letters*, 7(2):2116–2123, 2022.

[43] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[44] Yancheng Pan, Biao Gao, Jilin Mei, Sibo Geng, Chengkun Li, and Huijing Zhao. Semanticposs: A point cloud dataset with large quantity of dynamic instances. In *IEEE Intelligent Vehicles Symposium*, pages 687–693, 2020.

[45] Gilles Puy, Alexandre Boulch, and Renaud Marlet. Using a waffle iron for automotive point cloud semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3379–3389, 2023.

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[47] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[48] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer Assisted Intervention*, pages 234–241. Springer, 2015.

[49] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

 https://doi.org/10.52202/079017-4078

[50] Corentin Sautier, Gilles Puy, Spyros Gidaris, Alexandre Boulch, Andrei Bursuc, and Renaud Marlet. Image-to-lidar self-supervised distillation for autonomous driving data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9891–9901, 2022.

[51] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020.

[52] Suvrit Sra. A short note on parameter approximation for von mises-fisher distributions: and a fast implementation of i s (x). *Computational Statistics*, 27:177–190, 2012.

[53] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *European Conference on Computer Vision*, pages 685–702, 2020.

[54] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020.

[55] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019.

[56] Zhi Tian, Xiangxiang Chu, Xiaoming Wang, Xiaolin Wei, and Chunhua Shen. Fully convolutional one-stage 3d object detection on lidar range images. *Advances in Neural Information Processing Systems*, 35:34899–34911, 2022.

[57] Ozan Unal, Dengxin Dai, and Luc Van Gool. Scribble-supervised lidar semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2697–2707, 2022.

[58] Chengyao Wang, Li Jiang, Xiaoyang Wu, Zhuotao Tian, Bohao Peng, Hengshuang Zhao, and Jiaya Jia. Groupcontrast: Semantic-aware self-supervised representation learning for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[59] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2495–2504, 2020.

[60] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J. Kusner. Unsupervised point cloud pre-training via occlusion completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9782–9792, 2021.

[61] Ziyi Wang, Yongming Rao, Xumin Yu, Jie Zhou, and Jiwen Lu. Point-to-pixel prompting for point cloud analysis with pre-trained image models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(06):4381–4397, 2024.

[62] Lintai Wu, Qijian Zhang, Junhui Hou, and Yong Xu. Leveraging single-view images for unsupervised 3d point cloud completion. *IEEE Transactions on Multimedia*, pages 1–14, 2023.

[63] Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, and Shijian Lu. Transfer learning from synthetic to real lidar point cloud for semantic segmentation. In *AAAI Conference on Artificial Intelligence*, pages 2795–2803, 2022.

[64] Aoran Xiao, Jiaxing Huang, Weihao Xuan, Ruijie Ren, Kangcheng Liu, Dayan Guan, Abdulmotaleb El Saddik, Shijian Lu, and Eric Xing. 3d semantic segmentation in the wild: Learning generalized models for adverse-condition point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9382–9392, 2023.

[65] Binhui Xie, Shuang Li, Qingju Guo, Chi Liu, and Xinjing Cheng. Annotator: A generic active learning baseline for lidar semantic segmentation. *Advances in Neural Information Processing Systems*, 36, 2023.

[66] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *European Conference on Computer Vision*, pages 574–591, 2020.

[67] Jianyun Xu, Ruixiang Zhang, Jian Dou, Yushi Zhu, Jie Sun, and Shiliang Pu. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16024–16033, 2021.

[68] Mingye Xu, Mutian Xu, Tong He, Wanli Ouyang, Yali Wang, Xiaoguang Han, and Yu Qiao. Mm-3dscene: 3d scene understanding by customizing masked modeling with informative-preserved reconstruction and self-distilled consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4380–4390, 2023.

[69] Ryosuke Yamada, Hirokatsu Kataoka, Naoya Chiba, Yukiyasu Domae, and Tetsuya Ogata. Point cloud pre-training with natural 3d structures. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21283–21293, 2022.

[70] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[71] Junbo Yin, Dingfu Zhou, Liangjun Zhang, Jin Fang, Cheng-Zhong Xu, Jianbing Shen, and Wenguan Wang. Proposalcontrast: Unsupervised pre-training for lidar-based 3d object detection. In *European Conference on Computer Vision*, pages 17–33, 2022.

[72] 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.

[73] Sha Zhang, Jiajun Deng, Lei Bai, Houqiang Li, Wanli Ouyang, and Yanyong Zhang. Hvdistill: Transferring knowledge from images to point clouds via unsupervised hybrid-view distillation. *International Journal of Computer Vision*, pages 1–15, 2024.

[74] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9610, 2020.

[75] Yifan Zhang, Junhui Hou, and Yixuan Yuan. A comprehensive study of the robustness for lidar-based 3d object detectors against adversarial attacks. *International Journal of Computer Vision*, 132(5):1592–1624, 2024.

[76] Yifan Zhang, Siyu Ren, Junhui Hou, Jinjian Wu, and Guangming Shi. Self-supervised learning of lidar 3d point clouds via 2d-3d neural calibration. *arXiv preprint arXiv:2401.12452*, 2024.

[77] Yifan Zhang, Qijian Zhang, Junhui Hou, Yixuan Yuan, and Guoliang Xing. Unleash the potential of image branch for cross-modal 3d object detection. In *Advances in Neural Information Processing Systems*, volume 36, pages 51562–51583, 2023.

[78] Yifan Zhang, Qijian Zhang, Zhiyu Zhu, Junhui Hou, and Yixuan Yuan. Glenet: Boosting 3d object detectors with generative label uncertainty estimation. *International Journal of Computer Vision*, 131(12):3332–3352, 2023.

[79] Yifan Zhang, Zhiyu Zhu, Junhui Hou, and Dapeng Wu. Spatial-temporal graph enhanced detr towards multi-frame 3d object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[80] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10252–10263, 2021.

[81] Zhuoyang Zhang, Yuhao Dong, Yunze Liu, and Li Yi. Complete-to-partial 4d distillation for self-supervised point cloud sequence representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17661–17670, 2023.

[82] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.

[83] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9939–9948, 2021.

[84] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *Advances in Neural Information Processing Systems*, 36, 2023.

 https://doi.org/10.52202/079017-4078

A Appendix

In this appendix, we present the details omitted from the manuscript due to space constraints. The appendix is organized as follows:

- Section A.1: Theoretical justification for the application of the vMF distribution.
- Section A.2: Details of the dataset and evaluation metrics.
- Section A.3: Experimental setup of pre-training.
- Section A.4: More quantitative results.
- Section A.5: More qualitative results.
- Section A.6: Potential limitations of our method.
- Section A.7: Societal and environmental impact of our work.
- Section A.8: Public resources used in this work.

A.1 Theoretical Analysis

Proposition 1: The features of each class k can be modeled as a von Mises-Fisher (vMF) distribution. This means that for class k, the feature vectors g_i lie on a unit hypersphere and are centered around a mean direction μ_k with a concentration parameter κ_k.

Justification: To show that the features of each class can be effectively modeled by a vMF distribution, we use maximum likelihood estimation (MLE) to determine that the parameters μ_k and κ_k are optimal for the given set of feature vectors.

For a set of M_k feature vectors $\{g_i\}_{i=1}^{M_k}$ from class k, the likelihood function for the vMF distribution is:

$$L(\mu_k, \kappa_k) = \prod_{i=1}^{M_k} f(g_i; \mu_k, \kappa_k) = \prod_{i=1}^{M_k} \mathcal{K}_C(\kappa_k) \exp(\kappa_k \mu_k^T g_i) \tag{14}$$

Taking the natural logarithm of the likelihood function, we get the log-likelihood:

$$\log L(\mu_k, \kappa_k) = \sum_{i=1}^{M_k} \log f(g_i; \mu_k, \kappa_k) = M_k \log \mathcal{K}_C(\kappa_k) + \kappa_k \sum_{i=1}^{M_k} \mu_k^T g_i \tag{15}$$

Substituting the expression for $\mathcal{K}_C(\kappa_k)$, we get:

$$\log L(\mu_k, \kappa_k) = M_k \left[\log \left(\frac{\kappa_k^{C/2-1}}{(2\pi)^{C/2} I_{C/2-1}(\kappa_k)} \right) + \frac{\kappa_k}{M_k} \sum_{i=1}^{M_k} \mu_k^T g_i \right] \tag{16}$$

$$\log L(\mu_k, \kappa_k) = M_k \left[(C/2-1) \log \kappa_k - \log I_{C/2-1}(\kappa_k) - \frac{C}{2} \log(2\pi) + \frac{\kappa_k}{M_k} \sum_{i=1}^{M_k} \mu_k^T g_i \right] \tag{17}$$

To maximize the log-likelihood, we normalize μ_k by setting it to the normalized sum of the feature vectors:

$$\mu_k = \frac{\sum_{i=1}^{M_k} g_i}{\| \sum_{i=1}^{M_k} g_i \|} \tag{18}$$

The derivative of the log-likelihood with respect to κ_k is:

$$\frac{\partial \log L(\mu_k, \kappa_k)}{\partial \kappa_k} = M_k \left[\frac{C/2-1}{\kappa_k} - \frac{I_{C/2}(\kappa_k)}{I_{C/2-1}(\kappa_k)} + \frac{1}{M_k} \sum_{i=1}^{M_k} \mu_k^T g_i \right] \tag{19}$$

Setting this derivative to zero, we get:

$$\frac{C/2-1}{\kappa_k} - \frac{I_{C/2}(\kappa_k)}{I_{C/2-1}(\kappa_k)} + \frac{1}{M_k}\sum_{i=1}^{M_k} \mu_k^T g_i = 0 \tag{20}$$

Solving for κ_k, we obtain:

$$\kappa_k = \frac{\| \sum_{i=1}^{M_k} g_i \| (C - \| \sum_{i=1}^{M_k} g_i \|^2)}{1 - \| \sum_{i=1}^{M_k} g_i \|^2} \tag{21}$$

This equation allows us to compute the concentration parameter κ_k based on the alignment of the feature vectors. The concentration parameter κ_k is larger when the distribution is more tightly clustered around the mean direction, and smaller when the features are more uniformly spread across the hypersphere.

By maximizing the likelihood function for the vMF distribution, we have shown that the parameters μ_k and κ_k can be estimated to model the distribution of feature vectors for each class. The mean direction μ_k denotes the central direction of the feature cluster, and the concentration parameter κ_k controls the tightness of this clustering. Moreover, the way we estimate the parameters of vMF distribution in EMA is also consistent with the results of the above theoretical derivation.

Proposition 2: The representation of samples in the same class can vary significantly across different batches during contrastive distillation, and semantic-guided consistency regularization helps to learn structured features.

Justification: Without regularization, the representation of samples within the same class can vary significantly across different batches during contrastive distillation. This variance arises due to random sampling and the influence of negative samples in different batches. The weakly-supervised contrastive loss is defined as:

$$\mathcal{L}_{\text{sup}} = -\frac{1}{M_s}\sum_{i=1}^{M_s} \log \left[\frac{1}{|A(i)|} \sum_{a \in A(i)} \frac{\exp(\langle \boldsymbol{G}_i^{\text{3D}}, \boldsymbol{G}_a^{\text{2D}} \rangle / \tau)}{\sum_{j=1}^{M_s} \exp(\langle \boldsymbol{G}_i^{\text{3D}}, \boldsymbol{G}_j^{\text{2D}} \rangle / \tau)} \right], \tag{22}$$

The features of negative samples $\boldsymbol{G}_j^{\text{2D}}$ vary across batches, leading to different optimization paths for each mini-batch. This introduces variability in the learned representations $\boldsymbol{G}_i^{\text{3D}}$ for samples of the same class k.

When we do not use semantic-guided consistency regularization, the within-class variance for class k across different batches is:

$$\sigma_W^2 = \frac{1}{|B|}\sum_{B} \frac{1}{M_k}\sum_{i=1}^{M_k^B} \| g_i^k - \mu_k^B \|^2 \tag{23}$$

For ease of reading, we use g_i to refer to point feature $\boldsymbol{G}_i^{\text{3D}}$. And μ_k^B is the mean feature vector for class k in batch B. Due to the batch-wise variability in negative samples, μ_k^B can differ significantly across batches, leading to high within-class variance.

By minimizing the KL divergence, we align feature vectors g_i of class k with the mean direction μ_k, reducing the spread of feature vectors within the same class. The within-class variance with regularization is:

$$\sigma_W^2 = \frac{1}{K}\sum_{k=1}^{K} \frac{1}{M_k}\sum_{i=1}^{M_k} \| g_i^k - \mu_k \|^2 \tag{24}$$

Since μ_k is consistent across batches due to the regularization, the within-class variance is significantly reduced. This results in structured feature representations, enhancing class separability and improving performance in downstream tasks.

 https://doi.org/10.52202/079017-4078

Proposition 3: Learning structural representation during pretraining can benefit downstream tasks.

Justification: Structured features are those well-aligned within the same class (low within-class variance σ_W^2) and well-separated between different classes (high between-class variance σ_B^2).

With semantic-guided consistency regularization, feature vectors g_i^k for class k are closely aligned with the mean direction μ_k. This alignment reduces the within-class variance σ_W^2. Weakly-supervised contrastive learning pushes apart feature vectors of different classes, increasing the separation between class means μ_k. This increases the between-class variance σ_B^2.

Take the linear classifier as an example, the decision boundary is determined by the separation between class means. Higher σ_B^2 and lower σ_W^2 result in clearer decision boundaries, reducing classification errors.

Consider a simple linear classifier with weight vector w and bias b. The decision function is:

$$f(x) = w^{\mathrm{T}} x + b \tag{25}$$

The decision boundary is given by:

$$w^{\mathrm{T}} x + b = 0 \tag{26}$$

For well-structured features, the margin (distance between decision boundary and nearest samples) is maximized. The margin γ for class k can be expressed as:

$$\gamma = \frac{w^{\mathrm{T}}(\mu_k - \mu)}{\|w\|} \tag{27}$$

Higher between-class variance (σ_B^2) and lower within-class variance (σ_W^2) increase this margin, leading to better classification performance.

A.2 Dataset and Evaluation Metric

NuScenes Dataset. The NuScenes dataset, compiled from driving recordings in Boston and Singapore, utilizes a vehicle equipped with a 32-beam LiDAR and additional sensing technologies [6]. This comprehensive dataset is equipped with the typical sensor array found on autonomous vehicles, including a 32-beam LiDAR setup, six cameras, and radar systems, ensuring full 360-degree environmental perception. It includes 850 driving scene snippets, with 700 designated for training and 150 for validation, each scene lasting 20 seconds with annotations provided every 0.5 seconds. The dataset features extensive annotations across several object categories, including vehicles, pedestrians, bicycles, and road barriers, with each object encapsulated in a 3D bounding box and supplemented with attributes detailing visibility, activity, and pose.

NuScenes-lidarseg Dataset. The nuScenes dataset now encompasses features for semantic and panoptic segmentation through its extension, nuScenes-lidarseg [6]. This enhanced dataset provides semantic labeling across 32 distinct categories, with each point in the dataset's keyframes meticulously annotated. We utilize the 700 training scenes equipped with segmentation labels for refining our semantic segmentation models, and we assess model performance using the 150 scenes in the validation set.

SemanticKITTI Dataset. The SemanticKITTI (SK) dataset features paired RGB images and point cloud data derived from KITTI's urban scenes, specifically designed for semantic segmentation tasks [2]. This dataset is gathered using sensors mounted on a vehicle, including more than 200,000 images alongside their corresponding point clouds across 21 distinct sequences. Both images and point clouds are aligned to maintain a consistent relative transformation. Originally, the images are captured at a resolution of 1241x376 pixels, and each point cloud is composed of roughly 40,000 3D points. In line with standard practices, the dataset is divided into training and validation sets, with 10 sequences designated for training and the eighth sequence reserved for validation.

KITTI Dataset. KITTI is a crucial dataset for advancing 3D object detection in autonomous driving. With 7481 training and 7518 test point clouds, it covers diverse urban and suburban environments [20].

The dataset includes 3D point clouds and RGB images captured using a Velodyne HDL-64E LiDAR sensor. Calibration information between the camera and LiDAR is provided, essential for cross-modal knowledge transfer or sensor fusion tasks. Annotated with 3D bounding boxes, it features common objects like cars, pedestrians, and cyclists. The dataset is split into training (3712 samples) and validation (3769 samples) subsets.

ScribbleKITTI Dataset. ScribbleKITTI is derived from the SemanticKITTI dataset but introduces weak supervision in the form of line scribbles rather than fully labeled point clouds [57]. It retains the same set of 19,130 LiDAR scans, captured by a Velodyne HDL-64E sensor, but only about 8.06% of the semantic labels are provided compared to the fully-supervised SemanticKITTI dataset. This method of annotation drastically reduces the time required for labeling, offering around a 90% time saving. We use this dataset to evaluate how well models pre-trained on other datasets generalize under weaker annotations. For our experiments, we follow the SLidR protocol to create different splits of the training set, e.g., one scan is selected from every 100 frames to generate 1% labeled samples. The model's performance is evaluated on the official validation set.

RELLIS-3D Dataset. RELLIS-3D is a multimodal dataset collected from off-road environments on the Texas A&M University campus [25]. The dataset comprises 13,556 annotated LiDAR scans, providing a challenging scenario with complex terrain and class imbalance. It is valuable for assessing model performance in outdoor environments with varying topographies and object densities.

SemanticPOSS Dataset. SemanticPOSS is a smaller dataset focused on dynamic objects, captured on the campus of Peking University [44]. It includes 2,988 LiDAR scans from a Hesai Pandora 40-channel LiDAR sensor. The dataset is designed to challenge models with its focus on moving instances and dense environments, making it a useful resource for evaluating the adaptability of models to dynamic scenes. In our setup, sequences 00 and 01 are used to create half of the annotated training samples, and sequences 00 to 05, excluding sequence 02, are used for validation.

SemanticSTF Dataset. This dataset features 2,076 LiDAR scans collected under adverse weather conditions such as snow, fog, and rain, using a Velodyne HDL64 S3D sensor [64]. The dataset is split into training, validation, and test sets, ensuring an even distribution of weather conditions across all subsets. SemanticSTF is particularly suited for testing the robustness of models in extreme environmental conditions.

SynLiDAR Dataset. The SynLiDAR dataset is composed of synthetic point clouds generated in virtual environments using Unreal Engine 4 [63]. It consists of 13 sequences with a total of 198,396 scans. This synthetic dataset enables large-scale experimentation in scenarios that closely mimic real-world conditions, offering a controlled environment for model pre-training and testing. For our fine-tuning experiments, we use a uniformly downsampled subset of the dataset.

DAPS-3D Dataset. DAPS-3D includes both semi-synthetic and real-world data, with the subset DAPS-1 consisting of over 23,000 labeled LiDAR scans across 11 sequences [28]. The data was collected in the context of an autonomous robot's deployment in a real-world scenario. This dataset helps evaluate the transferability of models pre-trained on synthetic data to real-world tasks. In our setup, we use the sequence "38-18_7_72_90" for training and validate the model on the sequences "38-18_7_72_90", "42-48_10_78_90", and "44-18_11_15_32". This configuration helps in evaluating the model's performance on both synthetic and real-world data.

Robo3D (nuScenes-C) Benchmark. As a part of the Robo3D benchmark [30], nuScenes-C tests the robustness of models against various corruptions that simulate real-world challenges, such as severe weather and sensor malfunctions. These corruptions are categorized into different levels of severity (light, moderate, heavy), and the dataset includes eight types of disturbances like fog, snow, motion blur, and sensor interference. It is designed to assess how well models perform under out-of-distribution conditions. We follow the standard protocol of the Robo3D benchmark to evaluate model robustness under these out-of-distribution scenarios, using the official validation set to report results.

Evaluation Metrics. In semantic segmentation tasks, performance is assessed through Intersection-over-Union (IoU) for individual classes and mean IoU (mIoU) across all classes. In 3D object detection, the 3D detector's efficacy on the KITTI dataset is measured using Average Precision (AP) metrics at IoU thresholds of 0.7 for cars, 0.5 for pedestrians, and 0.5 for cyclists. Similarly, for the Waymo dataset, evaluation is based on 3D mean Average Precision (mAP).

 https://doi.org/10.52202/079017-4078

A.3 Experimental Setup of 3D Pretraining

Network Architectures. For the image processing branch, we utilize the ResNet-50 structure as the core architecture. This 2D backbone is initialized with weights that have been pre-trained using MoCov2 [10] on the ImageNet dataset. To preserve the receptive field while keeping the spatial resolution intact, we substitute the second and subsequent stridden convolutions with dilated convolutions, following established methodologies [50]. The up-sampling projection head includes a 1×1 convolutional layer that reduces the channel count from 2048 to 64, followed by a bi-linear interpolation up-sampling layer that enlarges the scale by a factor of 4. This up-sampling process effectively restores the resolution of the 2D feature map to match that of the original input images, specifically to the size of 416×224.

In the point cloud processing branch, we adopt two types of backbones. For the 3D semantic segmentation task, we employ the Sparse Residual 3D U-Net 34 (SR-UNet34) [48], adhering to practices previously established in SLidR [50]. The output from SR-UNet34 offers 256 channels, whereas the image branch outputs a 64-dimensional feature map. To align these dimensions, a 3D convolutional layer is used in the projection head to reduce the channel count of the point features to 64. We process the 3D point data into voxels to serve as input for the SR-UNet. The voxels are formatted in Cartesian coordinates covering an X-axis and Y-axis range of [-51.2m, 51.2m] and a Z-axis range of [-5.0m, 3.0m], with each voxel measuring (0.1m, 0.1m, 0.1m). To fully evaluate our method, we pre-train and transfer another VoxelNet [82] for the 3D object detection task.

Pre-training Details. We utilize momentum SGD for optimization, setting the initial learning rate at 0.5 and 0.01 for SR-UNet34 and VoxelNet respectively, with a momentum of 0.9 and a weight decay of 1e-4. To adjust the learning rate, we employ a cosine annealing scheduler [5] that gradually reduces it from the initial value to 0 over 50 epochs. The 3D network is pre-trained for these 50 epochs on four NVIDIA-3090 GPUs, processing a total batch size of 16, unless specified otherwise. For data augmentation, we incorporate several techniques. For the point cloud data, we apply random rotations around the z-axis, randomly flip the x and y axes, and omit points within a randomly selected cuboid, following the method described in [80]. For image data, augmentations include random horizontal flips and random crop-resize operations. In terms of generating weak semantic labels, the prompts we provided to the SEEM [84] encompass a total of 16 object categories: barrier, bicycle, bus, car, truck, trailer, motorcycle, construction vehicle, pedestrian, traffic cone, road, sidewalk, terrain, vegetation, building, and other ground. Unless otherwise specified, the semantic labels used in the ablation study are inferred with Grounded-SAM.

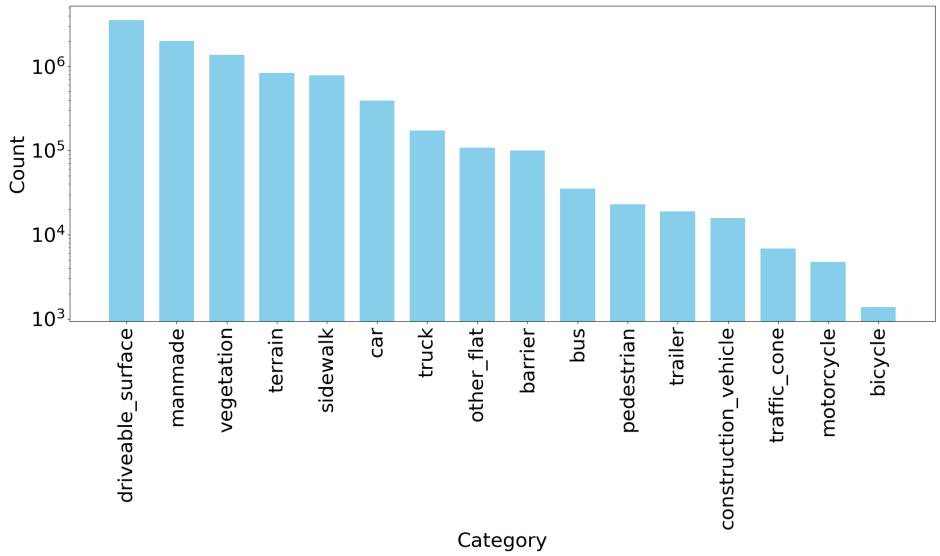

Figure 5: Class distribution at the pixel level for nuScenes dataset.

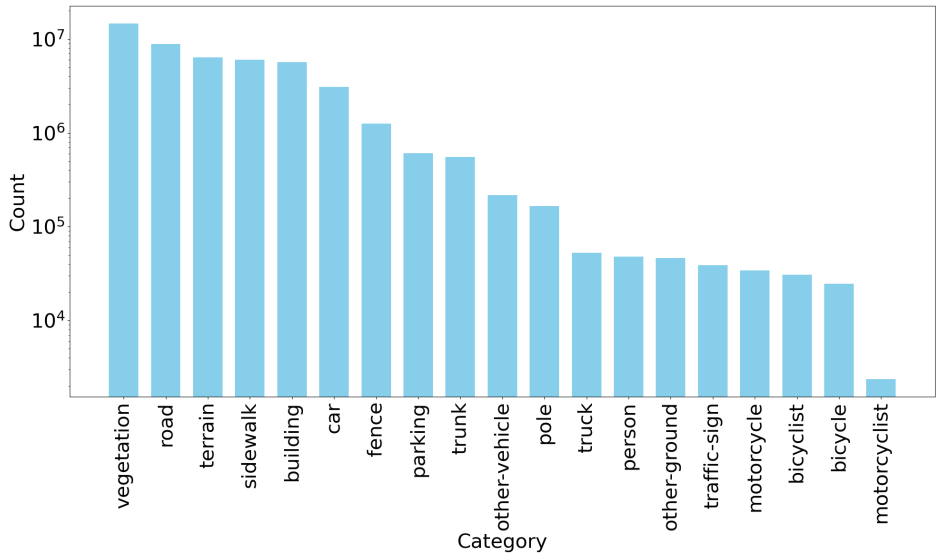

Figure 6: Class distribution at the pixel level for SemanticKITTI dataset.

A.4 More Quantitative Results

Effectiveness of Pre-training on A Stronger 3D Backbone. A key question arises: is the reduced effectiveness with larger data due to the model's insufficient size or inherent limitations in the proposed approach? When the available training data is limited, the benefits of pre-trained model weights on downstream tasks become more apparent. This phenomenon, commonly observed in self-supervised learning, is particularly important when downstream task data is scarce. In such cases, pre-trained representations offer a strong foundation, capturing crucial features that would otherwise remain unlearned from the limited labeled data. However, as the volume of labeled data increases, the model is capable of learning these features directly from the data itself, rendering the pre-trained representations less essential.

To further investigate this, we conducted experiments using a more robust 3D backbone, WaffleIron [45] (see Table 7). The results demonstrate that the effect of pre-training weights becomes less significant when sufficient training data is available for downstream tasks. This suggests that the reduced effectiveness with larger datasets is not due to the backbone's capability, but rather to the diminishing importance of pre-trained features as the model learns directly from ample labeled data.

Table 7: Performance for 3D backbone WaffleIron.

Method	1%	10%	100%
Random	33.26	58.13	77.60
Ours	50.14	66.43	78.21

Results on OOD Datasets. Table 8 presents the robustness evaluation of several state-of-the-art pretraining methods under eight out-of-distribution (OOD) corruption scenarios from the nuScenes-C dataset, part of the Robo3D benchmark. The corruptions include conditions such as fog, snow, motion blur, beam missing, cross-sensor interference, and more. The table reports three key metrics: mean Corruption Error (mCE, lower is better), mean Recovery Rate (mRR, higher is better), and mean Intersection over Union (mIoU, higher is better). Random Initialization: Among the randomly initialized models, Cylinder3D demonstrates the best overall performance in terms of mRR (78.08%) and mIoU (62.29%), indicating relatively better robustness across corruption scenarios. WaffleIron achieves the lowest mCE (106.73), but its performance in mRR and mIoU is slightly lower than Cylinder3D. SPVCNN also shows competitive results in mIoU (62.29%) but slightly lags in mRR. Pretrained Methods: Pretraining methods, such as PPKT, SLidR, and Seal, show a clear improvement in robustness compared to random initialization. Seal stands out with the second-best mCE (92.63) and mRR (83.08), along with a strong mIoU (72.66%). However, our method demonstrates the best

https://doi.org/10.52202/079017-4078

overall performance, achieving the lowest mCE (90.85) and the highest mRR (83.35), as well as competitive mIoU (70.62). This indicates that our approach outperforms both random initialization and other pretraining techniques in most OOD corruption scenarios, especially in beam missing (67.28%) and cross-sensor interference (59.47%). These results confirm that pretraining, particularly with our method, enhances the model's resilience to OOD corruptions, leading to more robust performance across varying environmental disturbances.

Table 8: Robustness evaluation of state-of-the-art pretraining methods under eight out-of-distribution corruptions in the *nuScenes-C* dataset from the Robo3D benchmark. All mCE (↓), mRR (↑), and mIoU (↑) scores are given as percentages (%).

Initial	Backbone	mCE (↓)	mRR (↑)	Fog	Wet	Snow	Motion	Beam	Cross	Echo	Sensor
Random	PolarNet [74]	115.09	76.34	58.23	69.91	64.82	44.60	61.91	40.77	53.64	42.01
Random	CENet [14]	112.79	76.04	67.01	69.87	61.64	58.31	49.97	60.89	53.31	24.78
Random	WaffleIron [45]	106.73	72.78	56.07	73.93	49.59	59.46	65.19	33.12	61.51	44.01
Random	Cylinder3D [83]	105.56	78.08	61.42	71.02	58.40	56.02	64.15	45.36	59.97	43.03
Random	SPVCNN [53]	106.65	74.70	59.01	72.46	41.08	58.36	65.36	36.83	62.29	49.21
Random	MinkUNet	112.20	72.57	62.96	70.65	55.48	51.71	62.01	31.56	59.64	39.41
PPKT	MinkUNet	105.64	76.06	64.01	72.18	59.08	57.17	63.88	36.34	60.59	39.57
SLidR	MinkUNet	106.08	75.99	65.41	72.31	56.01	56.07	62.87	41.94	61.16	38.90
Seal	MinkUNet	92.63	83.08	72.66	74.31	66.22	66.14	65.96	57.44	59.87	39.85
Ours	MinkUNet	90.85	83.35	70.62	75.86	66.51	64.06	67.28	59.47	62.90	47.94

Computational Cost. Our approach, OLIVINE, focuses on providing pre-trained weights and does not impact the inference speed of the model on downstream tasks. As seen in the Table 9, OLIVINE requires similar GPU memory and training time compared to other pre-training methods, demonstrating that our method does not significantly increase computational costs during pre-training.

Table 9: Comparison with other methods regarding the computational cost during pre-training.

Method	GPU Memory (GB)	Training Time (Hour)
PPKT	7.6	35.7
SLidR	10.7	38.9
OLIVINE	8.1	36.5

Class Unbalance in Datasets. The visualizations in Figures 5 and 6 illustrate the class distribution at the pixel level for the nuScenes and SemanticKITTI datasets, respectively. These figures reveal a significant class imbalance, a common challenge in many real-world datasets, where some classes are overwhelmingly more frequent than others. Such imbalance can skew the training process, leading to models that perform well on frequent classes but poorly on rare ones. This disparity predominantly affects the model's ability to generalize effectively across different scenarios, particularly underrepresented ones, resulting in biased predictions and reduced overall accuracy. For instance, infrequent but critical objects like pedestrians or bicycles might not be detected reliably, which is particularly concerning in autonomous driving contexts where safety is paramount.

To mitigate these issues, our method incorporates an optimized sampling strategy. This strategy involves adjusting the probability of selecting samples from underrepresented classes during the training process. By increasing the likelihood of including rare classes in the training set, we ensure that the model does not overlook these important but less frequent categories.

Per-class Performance. In Tables 10 and 11, we showcase the per-class performance of various point cloud pretraining strategies, including our method and other baselines, fine-tuned using just 1% of labeled data from the nuScenes-lidarseg and SemanticKITTI datasets. Our approach consistently surpasses other methods, achieving the highest mean Intersection over Union (mIoU) in nearly all categories. This marked superiority is also significant in complex categories like the bus and truck that demand more precise segmentation, highlighting our method's robustness in processing sparse and intricate data scenarios.

Effects of Different VFMs. The impact of different VFMs on the performance of OLIVINE is an important consideration. The precision of the semantic labels generated by these VFMs plays a crucial role in the success of OLIVINE. Those VFMs that also enable text prompts can be applied in OLIVINE to further improve its performance. Stronger VFMs are able to produce more accurate

Table 10: Per-class results on the nuScenes-lidarseg dataset using only 1% of the labeled data for fine-tuning. This chart displays the IoU scores for each category, with the highest and second-highest scores marked in dark blue and light blue, respectively.

Method	barrier	bicycle	bus	car	const. veh.	motor	pedestrian	traffic cone	trailer	truck	driv. surf.	other flat	sidewalk	terrain	manmade	vegetation	**mIoU**
Random	0.0	0.0	8.1	65.0	0.1	6.6	21.0	9.0	9.3	25.8	89.5	14.8	41.7	48.7	72.4	73.3	30.3
PointContrast	0.0	1.0	5.6	67.4	0.0	3.3	31.6	5.6	12.1	30.8	91.7	21.9	48.4	50.8	75.0	74.6	32.5
DepthContrast	0.0	0.6	6.5	64.7	0.2	5.1	29.0	9.5	12.1	29.9	90.3	17.8	44.4	49.5	73.5	74.0	31.7
PPKT	0.0	2.2	20.7	75.4	1.2	13.2	45.6	8.5	17.5	38.4	92.5	19.2	52.3	56.8	80.1	80.9	37.8
SLidR	0.0	1.8	15.4	73.1	1.9	19.9	47.2	17.1	14.5	34.5	92.0	27.1	53.6	61.0	79.8	82.3	38.3
ST-SLidR	0.0	2.7	16.0	74.5	3.2	25.4	50.9	20.0	17.7	40.2	92.0	30.7	54.2	61.1	80.5	82.9	40.8
Ours	0.0	12.8	74.3	82.9	13.5	43.1	58.3	31.2	20.9	47.6	93.6	40.2	59.8	66.1	81.9	82.6	50.5

Table 11: Per-class performance on SemanticKITTI with 1% of the labeled data utilized for fine-tuning. The figure shows the IoU for each class, where the top and second top scores are indicated by dark blue and light blue backgrounds, respectively.

Method	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign	**mIoU**
Random	91.2	0.0	9.4	8.0	10.7	21.2	0.0	0.0	89.4	21.4	73.0	1.1	85.3	41.1	84.9	50.1	71.4	55.4	37.6	39.5
PPKT	91.3	1.9	11.2	23.1	12.1	27.4	37.3	0.0	91.3	27.0	74.6	0.3	86.5	38.2	85.3	58.2	71.6	57.7	40.1	43.9
SLidR	92.2	3.0	17.0	22.4	14.3	36.0	22.1	0.0	91.3	30.0	74.7	0.2	87.7	41.2	85.0	58.5	70.4	58.3	42.4	44.6
Ours	93.1	17.5	28.1	45.2	18.7	47.4	31.4	0.0	91.8	32.3	75.5	1.8	88.1	47.2	85.7	59.0	71.4	59.8	43.9	49.4

semantic labels, which in turn lead to better learned representations during the pre-training process. As shown in the table below, the use of a stronger VFM, such as SEEM [84], can improve performance, highlighting the potential of our method when paired with more advanced models.

Table 12: Effects of different VFMs for generating semantic labels.

VFMs	LP	1%	5%	10%	25%
Grounded-SAM	47.30	46.12	57.51	63.04	69.39
Grounded-SAM-HQ	47.84	48.03	58.51	64.08	69.52
SEEM	**50.09**	**50.58**	**60.19**	**65.01**	**70.13**

A.5 More Qualitative Results

2D-3D Feature Similarities. The visualization presented in Figure 7 showcases the feature similarities between image and point cloud data as extracted through different projection heads. In the first column, the raw image is displayed alongside the projection of an anchor point within that image, setting the context for the comparison. The second column visualizes the feature similarities as extracted by the conventional projection heads, $h_{\text{3D}}^{\text{pp}}$ and $h_{\text{2D}}^{\text{pp}}$, used for point-pixel level contrastive distillation. Here, only the features of the pixel that directly correspond to the anchor point show a high degree of similarity, emphasizing a tight, point-to-point correspondence.

In contrast, the third column introduces results from the extra projection heads, $h_{\text{3D}}^{\text{sem}}$ and $h_{\text{2D}}^{\text{sem}}$, which are designed for weakly-supervised (category-aware) contrastive distillation. This setup reveals a broader similarity pattern, where points and pixels sharing the same category exhibit notably higher feature similarities. This suggests that our newly proposed projection heads are effective in capturing and reinforcing category-level feature associations across the 3D and 2D domains, thus enhancing the model's ability to recognize semantically similar but spatially disparate features.

Visual Results on Downstream Tasks. In Figures 8, 9, 10, and 11, we present additional qualitative results from fine-tuning tasks on downstream datasets. The application of pre-training strategies markedly improves model accuracy over baselines that use random initialization. Notably, our proposed OLIVINE outperforms SLiDR [50], highlighting its superior segmentation capabilities.

https://doi.org/10.52202/079017-4078

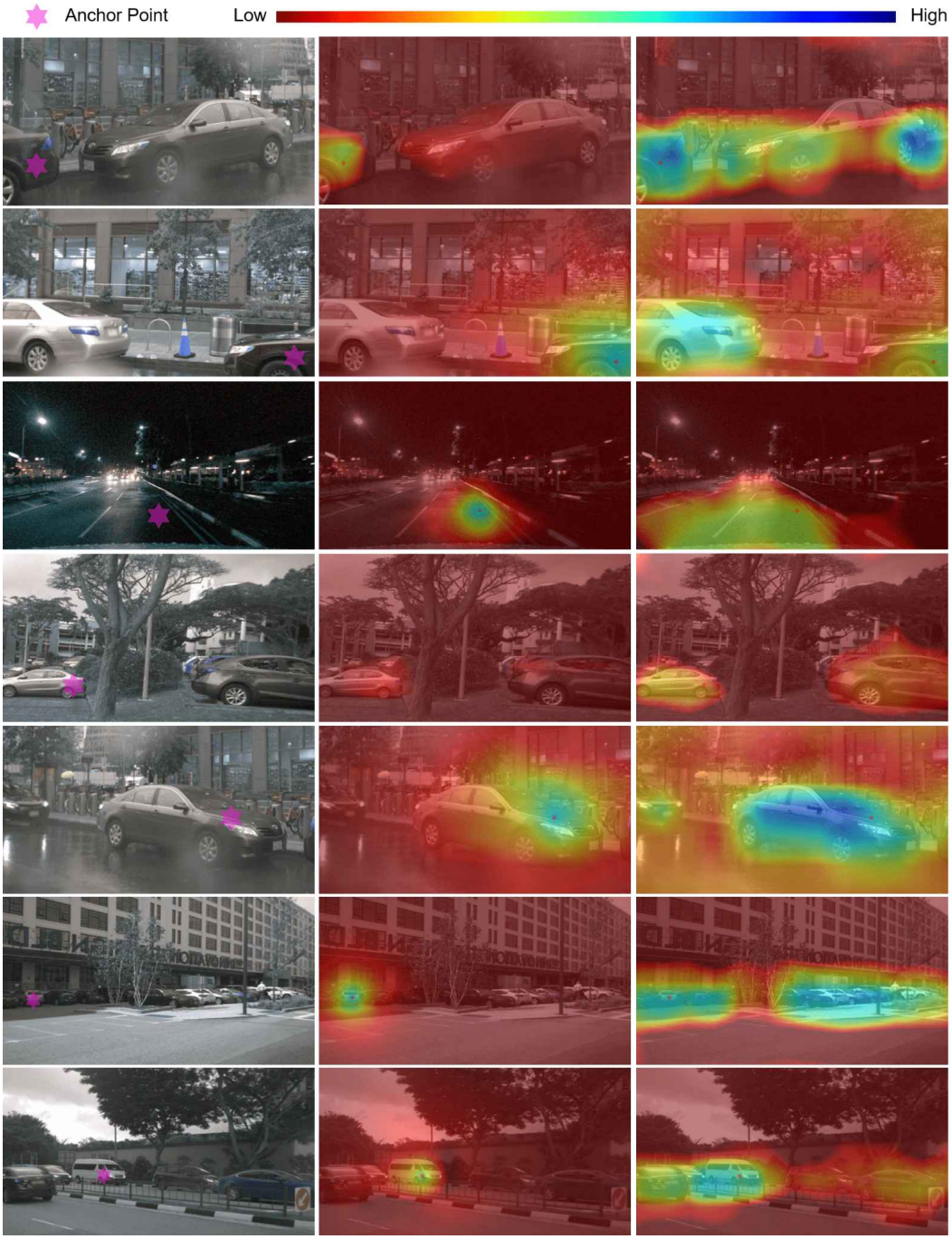

Figure 7: Visualization of the similarities between image and point cloud feature. In the first column, we show the raw image and the projection of anchor point in the image. In second columns, we illustrate the similarities between 3D query and 2D features extracted by the conventional projection heads $h_{\mathrm{3D}}^{\mathrm{pp}}$ and $h_{\mathrm{2D}}^{\mathrm{pp}}$ for point-pixel level contrastive distillation. In third columns, we illustrate the similarities between 3D query and 2D features extracted by the extra projection heads $h_{\mathrm{3D}}^{\mathrm{sem}}$ and $h_{\mathrm{2D}}^{\mathrm{sem}}$ for weakly-supervised (category-aware) contrastive distillation.

Despite these advancements, we note the occurrence of false positive predictions in edge cases, which we aim to address in future research.

Visualization of the Weak Semantic Labels. The weak labels generated by SEEM [84] using targeted prompts play a vital role in our processing pipeline. While reviewing these labels, we observe instances of imprecision (see Fig. 12 and Fig. 13). Should future advancements in Segmentation Anything Models yield more robust and accurate results, the effectiveness of our 3D pre-training strategy is likely to improve significantly. This progression would enhance our model's ability to interpret and learn from nuanced environmental data, ultimately leading to superior representation learning.

A.6 Potential Limitations

While our method, OLIVINE, effectively enhances the fine-grained image-to-LiDAR contrastive distillation process and demonstrates significant improvements in 3D scene understanding, there are technical and potential limitations that merit attention:

Semantic Label Accuracy. The accuracy of the weak semantic labels generated by VFMs is critical to the success of our model. Any deficiencies in these labels could propagate through the learning process, potentially compounding errors in the learned representations.

Training Data Diversity. Currently, our model is pre-trained using a single dataset, which may limit its applicability to environments or scenarios not well-represented in the training data. Expanding the training to include diverse datasets with varying characteristics could enhance the robustness and generalizability of our model.

Dependency on High-Quality Data Calibration. Our framework relies on the precise calibration and synchronization between LiDAR sensors and cameras. In real-world applications, perfect synchronization and calibration can be challenging to maintain, potentially affecting the accuracy and reliability of the semantic labels generated and subsequently the distillation process [50, 38, 73].

These limitations highlight areas for future development and research, suggesting a path toward more robust, adaptable, and efficient systems for 3D scene understanding in diverse and dynamic environments.

A.7 Societal and Environmental Impact

Our method, OLIVINE, which enhances image-to-LiDAR contrastive distillation using Visual Foundation Models, significantly impacts society and the environment. Societally, it boosts the safety and reliability of autonomous systems, increasing public trust and improving data analysis in various industries. Environmentally, deploying advanced deep learning models requires increased computational resource usage, which can lead to higher energy consumption and associated carbon emissions. This is particularly relevant during the intensive training phases that require high-performance GPUs and long training durations, especially as the model scales to larger datasets or more complex scenarios. Conversely, by distributing our pre-trained models, we aim to reduce the need for repetitive training across multiple downstream tasks, which can decrease the overall computational load and energy consumption needed to achieve high performance in various applications. This aspect potentially mitigates some of the environmental costs.

A.8 Public Resources Used

We acknowledge the use of the following public resources, during the course of this work:

- Grounded-Segment-Anything[2] ..... Apache License 2.0
- KITTI Dataset[3] ..... CC BY-NC-SA 3.0
- MinkowskiEngine[4] ..... MIT License
- nuScenes[5] ..... CC BY-NC-SA 4.0

[2] https://github.com/IDEA-Research/Grounded-Segment-Anything.
[3] https://www.cvlibs.net/datasets/kitti.
[4] https://github.com/NVIDIA/MinkowskiEngine.
[5] https://www.nuscenes.org/nuscenes.

https://doi.org/10.52202/079017-4078

- ScribbleKITTI[6] ..... Unknown
- RELLIS-3D[7] ..... CC BY-NC-SA 3.0
- SemanticPOSS[8] ..... Unknown
- SemanticSTF[9] ..... CC BY-NC-SA 4.0
- SynLiDAR[10] ..... MIT License
- DAPS-3D[11] ..... MIT License
- nuScenes-C[12] ..... CC BY-NC-SA 4.0
- nuScenes-devkit[13] ..... Apache License 2.0
- OpenPCDet[14] ..... Apache License 2.0
- PyTorch-Lightning[15] ..... Apache License 2.0
- SemanticKITTI[16] ..... CC BY-NC-SA 4.0
- SLidR[17] ..... Apache License 2.0
- SEEM[18] ..... Apache License 2.0

[6] https://github.com/ouenal/scribblekitti.
[7] http://www.unmannedlab.org/research/RELLIS-3D.
[8] http://www.poss.pku.edu.cn/semanticposs.html.
[9] https://github.com/xiaoaoran/SemanticSTF.
[10] https://github.com/xiaoaoran/SynLiDAR.
[11] https://github.com/subake/DAPS3D.
[12] https://github.com/ldkong1205/Robo3D.
[13] https://github.com/nutonomy/nuscenes-devkit.
[14] https://github.com/open-mmlab/OpenPCDet.
[15] https://github.com/Lightning-AI/lightning.
[16] http://semantic-kitti.org.
[17] https://github.com/valeoai/SLidR.
[18] https://github.com/UX-Decoder/Segment-Everything-Everywhere-All-At-Once.

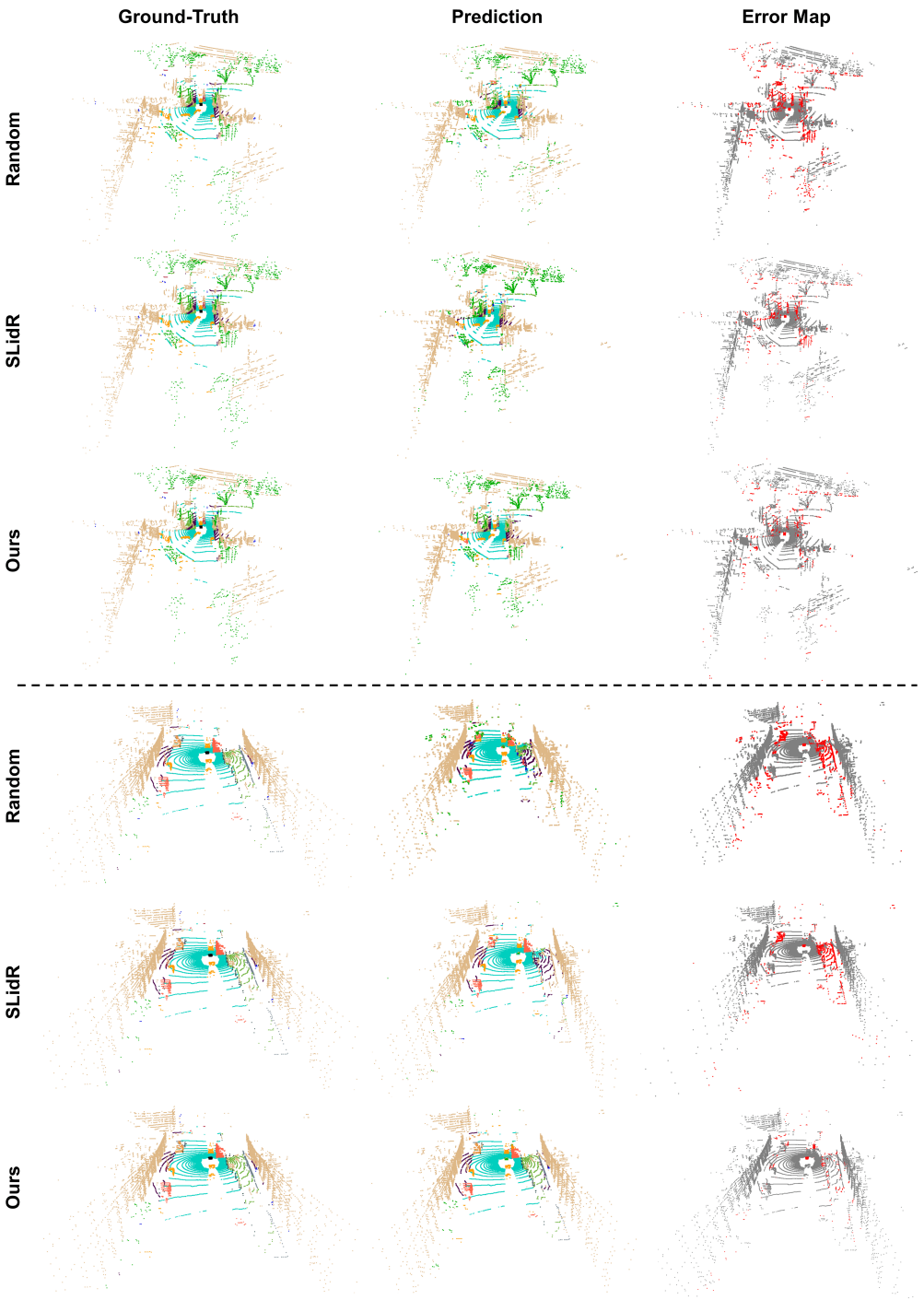

Figure 8: Qualitative results of fine-tuning on 1% of the nuScenes-lidarseg dataset with different pre-training strategies. Note that the results are shown as error maps on the right, where red points indicate incorrect predictions. Best viewed in color and zoom in for more details.

https://doi.org/10.52202/079017-4078

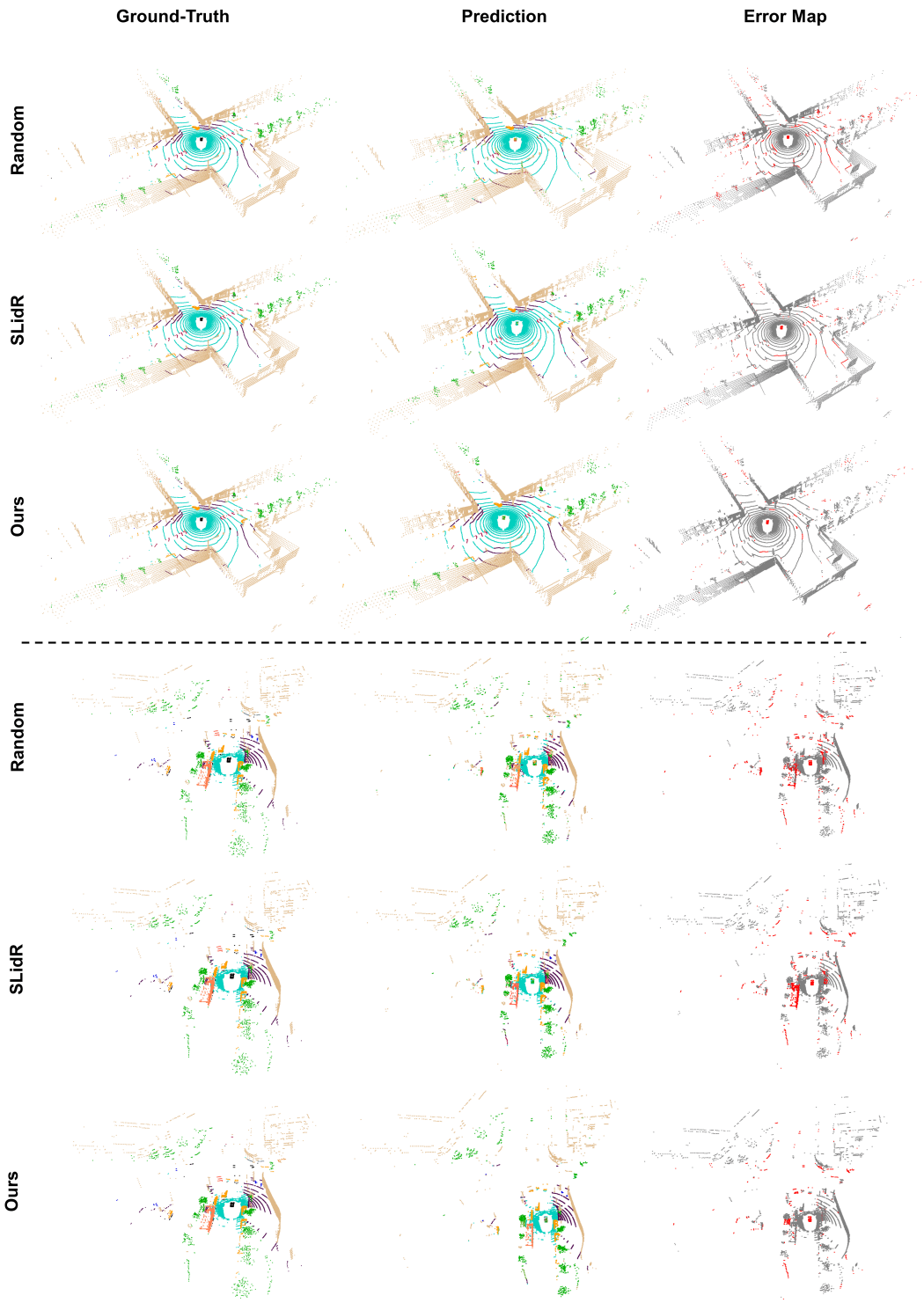

Figure 9: Qualitative results of fine-tuning on 1% of the nuScenes-lidarseg dataset with different pre-training strategies. Note that the results are shown as error maps on the right, where red points indicate incorrect predictions. Best viewed in color and zoom in for more details.

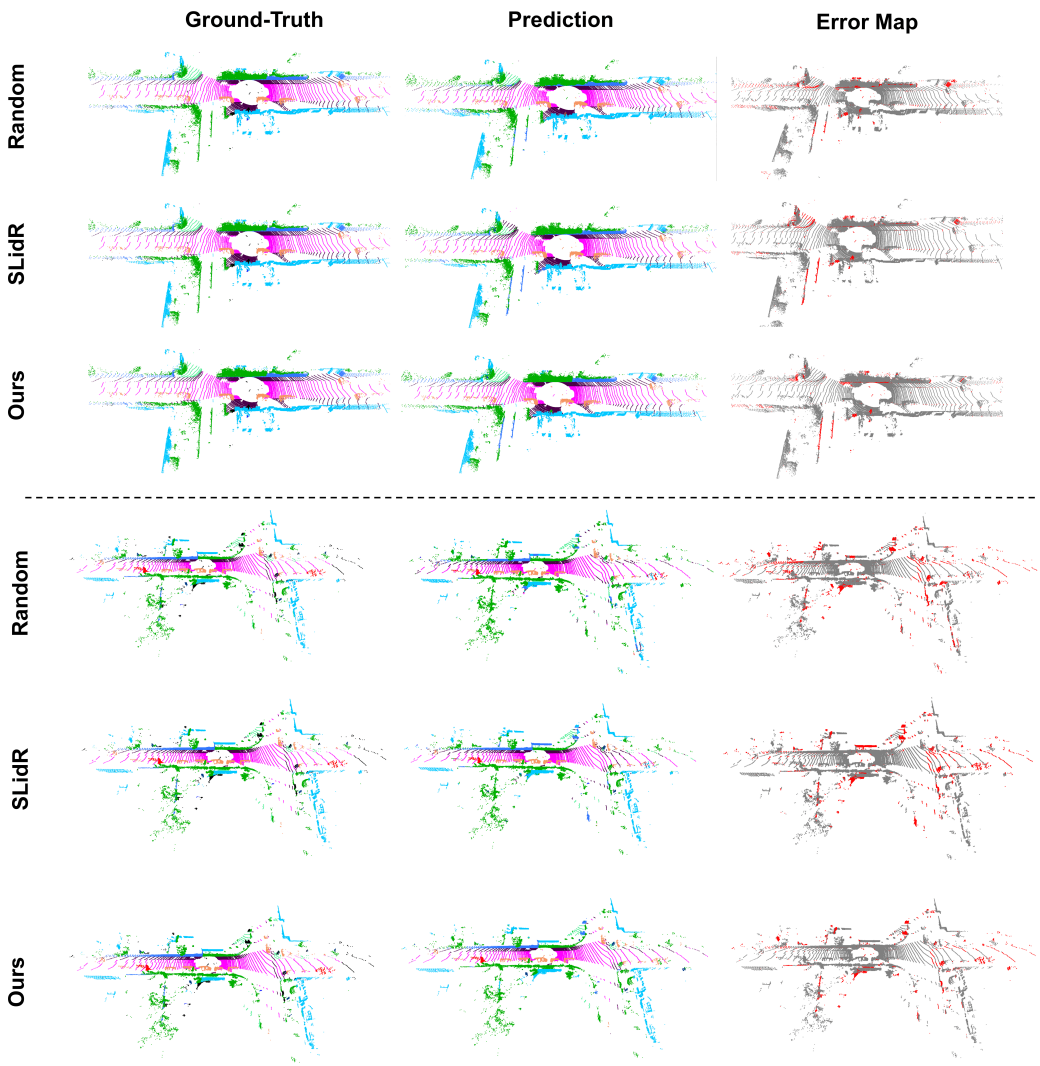

Figure 10: Qualitative results of fine-tuning on 1% of the SemanticKITTI dataset with different pre-training strategies. Note that the results are shown as error maps on the right, where red points indicate incorrect predictions. Best viewed in color and zoom in for more details.

https://doi.org/10.52202/079017-4078

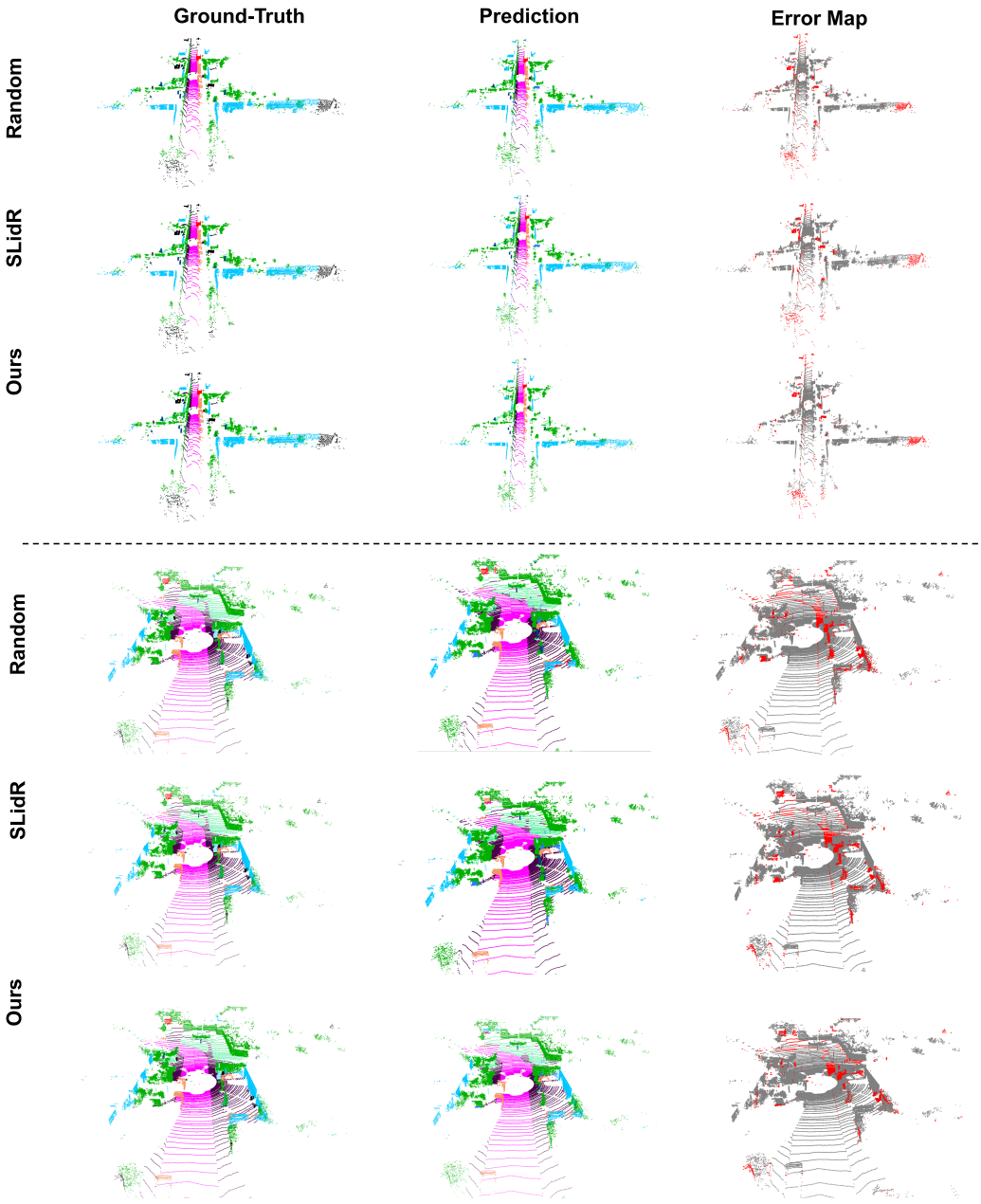

Figure 11: Qualitative results of fine-tuning on 1% of the SemanticKITTI dataset with different pre-training strategies. Note that the results are shown as error maps on the right, where red points indicate incorrect predictions. Best viewed in color and zoom in for more details.

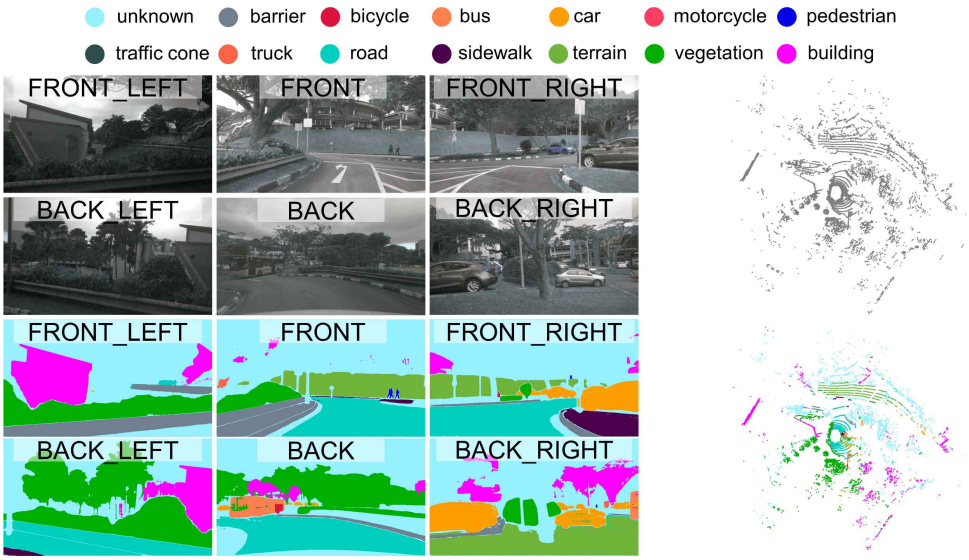

Figure 12: Illustration of the weak semantic labels predicted by Grounded SAM. The top half of the figure displays the raw RGB images and LiDAR point clouds, while the bottom half presents the corresponding weak semantic labels applied to both images and point clouds, aligned using camera parameters. Each distinct segment is represented by a unique color. Best viewed in color.

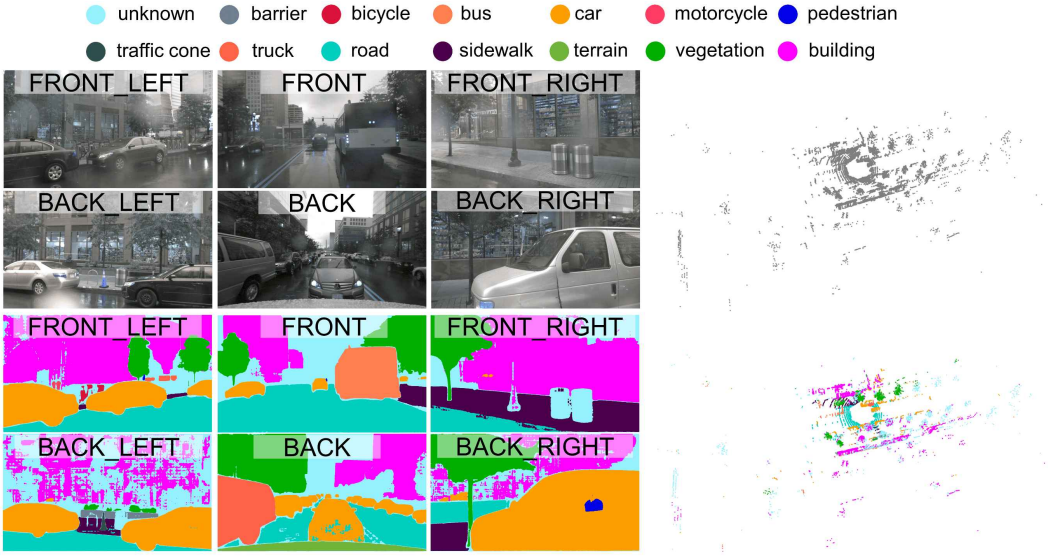

Figure 13: Illustration of the weak semantic labels predicted by Grounded-SAM. The top half of the figure displays the raw RGB images and LiDAR point clouds, while the bottom half presents the corresponding weak semantic labels applied to both images and point clouds, aligned using camera parameters. Each distinct segment is represented by a unique color. Best viewed in color.

https://doi.org/10.52202/079017-4078

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope. These sections effectively summarize the key aspects of the research. They set clear expectations for the reader about the methodologies employed and the advancements over existing techniques.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the potential limitations of our method in Section A.6.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The focus of our paper is primarily on the application and empirical validation of VFMs for image-to-LiDAR contrastive distillation, rather than on deriving new theoretical results. Consequently, our work does not introduce formal theorems or proofs that would necessitate a detailed discussion of assumptions or a mathematical proof structure.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our paper provides detailed methodology and experimental settings necessary to reproduce the main results, supporting our core claims and conclusions.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We attach the code in supplementary materials, where we provide sufficient instructions to reproduce the main results. And we will release the source code on github in the future.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper meticulously details all training and testing parameters, including data splits, hyperparameters, selection criteria, and the type of optimizer used, ensuring complete understanding of the experimental results.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are not included due to the computational expense involved. However, we generally ensure robustness against random seed variations by reporting results averaged over three independent experiments. For the fine-tuning on 3D detection task, we recorded the best results from the three experiments.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer to the details in Section A.3.

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Yes, we adhere to the NeurIPS Code of Ethics.

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the societal impacts of our paper in Section A.7.

11. **Safeguards**

https://doi.org/10.52202/079017-4078

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our paper does not involve the release of models or datasets that pose a high risk for misuse. Therefore, safeguards for responsible release and use were not necessary for this research. We will consider this issue when we release our pre-trained model.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the relevant papers and credit the term of used resource in Section A.8.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We attach the source with documentation in the supplementary materials anonymously.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: the paper does not involve crowdsourcing nor research with human subjects.