SELF-SUPERVISED REPRESENTATION LEARNING FROM LIDAR IMAGES

Anonymous authors
Paper under double-blind review

ABSTRACT

In real-world applications, such as autonomous driving, the inherent complexity of the environment poses high requirements on the quality of the representations learned by perception models. While semantic understanding is important, it is usually not present natively in Light Detection and Ranging (LiDAR) data. To solve this, for the first time, we investigate self-supervised learning for LiDAR range-view representation with the hypothesis that distillation from Vision Foundation Models (VFMs) fed with LiDAR images can be beneficial. The motivation is three-fold: datasets with annotated LiDAR data are rather scarce since they are costly and time-consuming to obtain, the rapid development of LiDAR technology makes existing datasets obsolete quickly, and perception solutions face the challenges of representation learning for complex environments in real world applications of autonomous driving. We pose two research questions: Can the semantic knowledge learned by VFMs from natural RGB images be transferred to LiDAR representations even when no RGB/camera data are available during the LiDAR pretraining process? And if so, what knowledge is actually transferred across this large modality gap? After full fine-tuning, Self-Supervised Learning (SSL) initialization matches the downstream semantic segmentation score of an ImageNet-21k-initialized supervised baseline with the same architecture, and VFM distillation yields substantially higher linear-probe scores than Masked Auto-Encoding (MAE). Our results reveal potential in LiDAR image-based self-supervised representation learning for unlabeled long-range, high-resolution sensors, paving the way for scalable learning for modern LiDAR systems. Our code will be publicly released on GitHub after the paper is accepted for publication.

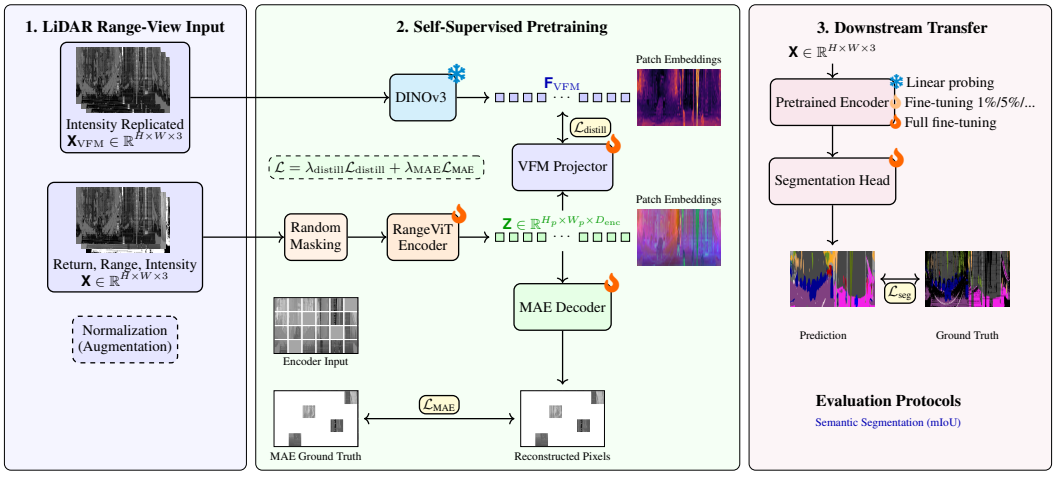

Figure 1: Overview of our Self-Supervised Learning (SSL) and evaluation framework. (1) Preparation of Light Detection and Ranging (LiDAR) images to be fed to the Vision Foundation Model (VFM) and our encoder. (2) Self-supervised pre-training, consisting of VFM distillation and Masked Auto-Encoding (MAE). (3) Evaluation of Downstream Transfer on a semantic segmentation task.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Learning representations that capture an informative scene understanding is a fundamental problem in autonomous driving. Traditionally, most research has focused on perception for specific tasks, such as semantic segmentation or object detection. Meanwhile, recent research shows how a strong learned representation can be useful for virtually any downstream tasks. We follow the latter mindset and investigate encoding LiDAR data in a representation as rich as possible, that may later be processed by lightweight decoders for specific downstream tasks.

LiDAR range-view images are a compact representation of LiDAR data, typically including range and intensity channels, in the image plane; see Figure 2. For most LiDAR sensors, a projection from the 3D space to the 2D image plane is required to achieve this representation, but some modern sensors provide this representation natively. An advantage of this input representation is that simple image processing neural network architectures can be used instead of more complex, 3D point-based models that implement specific serialization (Wu et al., 2024) or voxel-based models that implement operators engineered to handle the huge sparsity in the 3D space (Graham et al., 2018). Additionally, a number of advances in image processing can be leveraged, such as VFM distillation or pre-trained weight initialization. Furthermore, in the range-view, all data regardless of their 3D position is included, contrary to 3D grid-based layouts where often a grid limit is enforced to keep computations bounded.

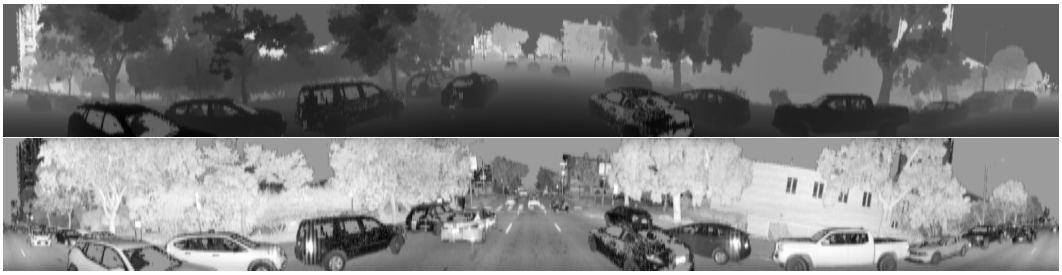

Figure 2: An image from an Ouster Rev 8 LiDAR. Channels are range (top) and intensity (bottom).

As the quality and resolution of LiDARs continues to improve dramatically over time, more datasets contain higher-quality, higher-resolution LiDAR data (Li et al., 2021; Jung et al., 2024; Ouster Inc, 2026). But given the difficulty and cost of annotating 3D data, most do not contain the required labels for them to be used in supervised learning for perception tasks. Even when labels are available, they are not necessarily accurate (Justo Miro et al., 2026). Furthermore, the available LiDAR data is only a fraction of the available camera image data. Therefore, SSL can help to extract rich representations from such rich data when labels are not available. Notably, SSL enables data scaling, which is a primary factor in model performance (Li et al., 2026).

While modern VFMs have shown remarkable semantic understanding from large-scale image data, transferring such knowledge to LiDAR sensors remains an open challenge. We explore representation learning through an SSL framework that grounds learning in sensor observations through VFM supervision and geometric reconstruction objectives, enabling us to learn rich LiDAR representations from unlabeled LiDAR images.

Our work lies in the intersection between LiDAR image-based perception and SSL. We make the following novel contributions:

- Camera-free cross-modal knowledge transfer: we describe a method for transferring directly to LiDAR images the representations learned by an RGB VFM, without requiring camera images, camera-LiDAR calibration, or cross-modal correspondences.
- Comparison of semantic and geometric SSL objectives: VFM distillation is expected to provide semantic/high-level knowledge, while MAE is expected to provide LiDAR-native geometric and structural learning. We find that these two objectives are useful on their own, but VFM distillation is significantly stronger.
- Representation scalability and transferability: we show that the learned representations are useful in downstream evaluation across label-scarce settings and datasets/sensors.

2 RELATED WORK

Perception on LiDAR images has primarily focused on semantic segmentation. Early works such as SqueezeSeg (Wu et al., 2018; 2019), RangeNet++ (Milioto et al., 2019), and SalsaNext (Cortinhal et al., 2020) demonstrated that range-view projections enable efficient semantic segmentation using image-based architectures, while RangeViT (Ando et al., 2023) introduced Vision Transformer (ViT) for LiDAR image understanding. More recently, methods such as RangeFormer (Kong et al., 2023) and RangeSAM (Kühn et al., 2026) have leveraged transformer architectures and VFMs to further improve segmentation quality. Range-view representations have also been successfully applied to 3D object detection (Meyer et al., 2019; Liang et al., 2021; Bai et al., 2023) and temporal tasks (Meyer et al., 2020; Mersch et al., 2022). However, these approaches are predominantly trained in a fully supervised manner on labeled datasets.

A line of research has investigated SSL for point clouds (Fei et al., 2026) and autonomous driving. Regarding object-level point clouds, Wang et al. (2022) convert the 3D data into colored images, which are then processed by a pre-trained, frozen image model. Zhang et al. (2023) convert object-level point clouds into multi-view images, which are used as a guide for 3D masking. BlendCLIP (Khoche et al., 2026) infuses categorization knowledge from object-level datasets into point clouds from autonomous driving datasets. Sonata (Wu et al., 2025) pre-trains via SSL a PTv3 backbone on RGB-D data from indoor datasets, achieving rich semantics and adapting very well to outdoor datasets for semantic segmentation.

In the field of cross-modal learning, RangeLVDet (Zhang et al., 2021) fuses features from multiple LiDAR input representations and camera RGB data. SuperFlow (Xu et al., 2024b) projects LiDAR onto multi-view cameras and feeds camera to VFM DINOv2 and LiDAR to MinkUNet. It does spatial contrastive learning between each LiDAR-camera pair and temporal contrastive learning among consecutive LiDAR point clouds across scenes. KDRangeViT (He et al., 2025) does distillation from camera ViT into range images. CleverDistiller (Govindarajan et al., 2025), and GASP (Ljungbergh et al., 2026) distill knowledge from camera images into 3D LiDAR data through VFMs. Concerto (Zhang et al., 2025) evolves Sonata by leveraging information from 2D and 3D via joint embedding prediction. HilDA (Wozniak et al., 2026) combines VFM knowledge distillation from camera images into LiDAR data with 3D temporal pre-text tasks to build robust representations in the 3D voxel space. xModel-KD (Pathmanathan et al., 2026) makes a cross-modal (camera, LiDAR) encoder where LiDAR and camera data are fed to separate encoders (a 3D and a 2D one, respectively) and their features are applied a contrastive loss and later decoded into 3D and 2D segmentation, respectively.

VFMs are trained on internet-scale amounts of image data in a self-supervised manner. The size of such data is orders of magnitude larger than the size of available LiDAR data. This makes VFMs a rich source of information. Some recent examples of such models are DINOv2 (Oquab et al., 2023), DINOv3 (Siméoni et al., 2025). To this end, a line of research focuses on distilling knowledge from VFMs into scene-level LiDAR data (Ljungbergh et al., 2026; Wozniak et al., 2026). Typically, this is done by inferring VFM features on camera images, and pairing camera pixels and LiDAR points through spatial projections to then distill VFM pixel features into LiDAR points. TOLiD (Mahendran et al., 2026) lifts Bird's-Eye View (BEV) tokens to the ViT space to be able to distill from VFM. While these prior works have shown excellent results, its applicability is limited to datasets that contain both camera and LiDAR data.

MAE originated as a SSL technique for images (He et al., 2022), where it was concluded that full token masking at a ratio of 75% both made powerful representations emerge and training be faster. Then came masking on 3D object-level point clouds (Yu et al., 2022) and on scene-level LiDAR (Yang et al., 2023; Krispel et al., 2024). PICTURE (Xu et al., 2024a) highlights the need of high-level feature supervision to complement or remove low-level reconstruction, and uses Seal (VFM) features and a class-based masking as reconstruction target.

Most existing SSL approaches obtain VFM semantics from RGB images and establish camera–LiDAR correspondences to transfer knowledge cross-modality. Our approach instead investigates whether a VFM, complemented by MAE, can provide useful representation-level supervision directly from range-view LiDAR images, without camera-LiDAR correspondences, sensor-to-sensor calibration, or paired RGB observations.

3 METHODOLOGY

We provide an overview of our method in Figure 1, which is detailed in the following sections.

3.1 INPUT DATA PREPARATION

For the datasets that do not provide LiDAR data in image format natively, we project the 3D points to the image plane by mapping the spherical coordinates to horizontal and vertical pixel indices (Ando et al., 2023). Let the input image for valid return mask, $\mathbf{X}^m \in \{0,1\}^{H\times W}$; range (depth), $\mathbf{X}^d \in \mathbb{R}^{H\times W}$; and intensity, $\mathbf{X}^i \in \mathbb{R}^{H\times W}$.

Multiple pixels in the range-view images are empty due to the laser not getting any light reflected back or projection artifacts; we set these pixel values to zero in all three channels. Further, we apply normalization to the images as follows.

The range in LiDAR data is highly long-tailed, so to normalize it we apply the logarithm normalization from equation 1 to it,

$$\tilde{\mathbf{X}}^d = \log_e(1 + \mathbf{X}^d). \tag{1}$$

This results in a positive, more concentrated distribution.

Typically, the RGB data in datasets, thus the data that is fed to models, are not raw sensor measurements. Instead, raw sensor measurements undergo Image Signal Processing (ISP) pipelines that transform the sensor's linear measure of light photons into perceptually pleasing, sRGB images (Ljungbergh et al., 2023). As the intensity in LiDAR is a measure of the strength of the returned (backscattered) laser signal and is approximately proportional to the amount of received optical energy or photons (Kashani et al., 2015), we apply a similar normalization to the LiDAR intensity. The ISP that we apply is minimal and consists of saturating the 1st and 99th percentiles to reduce the over-influence of outliers (e.g. extremely bright objects such as license plates or traffic signs), and subsequently applying a rescaling to $[0, 1]$ and a standard sRGB nonlinear encoding (opto-electronic transfer function) defined in IEC 61966-2-1 International Electrotechnical Commission (1999) to each linear component as per equation 2

$$\tilde{\mathbf{X}}^i_{h,w} = \begin{cases} 12.92 \cdot \mathbf{X}^i_{h,w}, & \mathbf{X}^i_{h,w} \leq 0.0031308, \\ 1.055 \cdot \left(\mathbf{X}^i_{h,w}\right)^{1/2.4} - 0.055, & \mathbf{X}^i_{h,w} > 0.0031308; \end{cases} \tag{2}$$

where h and w are the height and width pixel indices. This results in a VFM-friendly, RGB-like distribution.

As input values with average close to zero and similar covariances are known to make training stable (LeCun et al., 2002), we apply standardization transforms. We use the mean, μ, and standard deviation, σ, as specified in Section A.2, and apply normalization

$$N(\tilde{\mathbf{X}}) = (\tilde{\mathbf{X}} - \mu)/\sigma. \tag{3}$$

The expected encoder input is a LiDAR image, which can be of any resolution that is a multiple of the configurable patch size. Each LiDAR image can have an arbitrary number of channels, which we set to 3: valid return mask (so that the model has a cue for which pixels are actual zeros and which ones are no-returns), range, and intensity. In turn, each LiDAR image that we input to the VFM must be RGB-like, meaning having 3 channels. We use the intensity channel replicated 3 times, as this resembles RGB more than range, which measures distance, another physical quantity. Let the input image for the encoder, $\mathbf{X} = \text{Concatenate}\left(\mathbf{X}^m, N\left(\tilde{\mathbf{X}}^d\right), N\left(\tilde{\mathbf{X}}^i\right)\right) \in \mathbb{R}^{H\times W\times 3}$; and for the VFM, $\mathbf{X}_{\text{VFM}} = \text{Concatenate}\left(N\left(\tilde{\mathbf{X}}^i\right), N\left(\tilde{\mathbf{X}}^i\right), N\left(\tilde{\mathbf{X}}^i\right)\right) \in \mathbb{R}^{H\times W\times 3}$.

As both our encoder and the VFM teacher are ViT, we must choose image and patch resolution carefully. The RangeViT architecture that we use as our encoder is flexible enough to be configured with any patch size, even non-square ones. We note that the aspect ratio of LiDAR images is typically way wider than that of camera images. Thus, we choose a patch size of 2×8, which was shown to work best on the patch size ablation study of Ando et al. (2023). Since our teacher

VFM is DINOv3, whose pretrained weights enforce a patch size of 16×16, we bilinearly upsample the input image fed to it such that each patch in the teacher corresponds exactly to a patch in the encoder. For example, for an image of 64×2048 and an encoder patch size of 2×8, we upsample the VFM input by a factor $16/2 = 8$ vertically and $16/8 = 2$ horizontally, into a resulting image resolution of 512×4096.

3.2 ENCODER

We build upon RangeViT's (Ando et al., 2023) encoder of LiDAR images. This encoder is made of a patch embedding stem that is tailored for LiDAR images, and a ViT, and at its output provides patch tokens and a CLS token. We use the smallest version, which has embedding dimension $D_{\text{enc}} = 384$. Our LiDAR image encoder produces a representation in the form of high-dimensional features, $\mathbf{Z}$, as per equation 4,

$$\mathbf{Z} = E(\mathbf{X}; \boldsymbol{\theta}_E), \qquad \mathbf{Z} \in \mathbb{R}^{(1+H_p W_p) \times D_{\text{enc}}}, \tag{4}$$

where $\boldsymbol{\theta}_E$ are the encoder's trainable parameters, and the flattened dimensions $1 + H_p W_p$ correspond to 1 CLS token and $H_p W_p$ patch tokens.

3.3 SELF-SUPERVISED PRE-TEXT TASKS

We design two pre-text tasks. First, we use a VFM teacher with pre-trained weights, that we keep frozen, to provide patch and CLS embeddings, $\mathbf{F}_{\text{VFM}} = \text{VFM}(\mathbf{X}_{\text{VFM}}) \in \mathbb{R}^{(1+H_p W_p) \times D_{\text{VFM}}}$, that our encoder uses as supervision signal. For compute efficiency, we generate these once and save them to disk so that the VFM does not occupy GPU memory during training. While any ViT-based VFM is compatible with our framework, we use DINOv3 (Siméoni et al., 2025), variant ViT-7B16. This VFM has an embedding dimension $D_{\text{VFM}} = 4096$ and a patch size 16×16.

By feeding channels of the same LiDAR sample to both our encoder and the VFM, we achieve a perfect one-to-one matching between the patch features of both. This avoids issues that arise when using the VFM on camera images and relying on LiDAR-to-camera spatial projections, such as mismatching fields of view between sensors, or dependency on perfect sensor-to-sensor calibration, which has been discussed as limitations in prior art (Ljungbergh et al., 2026; Wozniak et al., 2026).

We distill both from patch features and the CLS feature. As the VFM's embedding dimension differs from that of our encoder, we attach at the end of our encoder a lightweight linear trainable layer that projects our encoder's features onto the VFM's embedding space as per equation 5,

$$\mathbf{F} = \mathbf{Z}\boldsymbol{W}_{\text{VFM}} + \boldsymbol{b}_{\text{VFM}}, \qquad \mathbf{F} \in \mathbb{R}^{(1+H_p W_p) \times D_{\text{VFM}}}, \tag{5}$$

where $\boldsymbol{W}_{\text{VFM}} \in \mathbb{R}^{D_{\text{enc}} \times D_{\text{VFM}}}$ and $\boldsymbol{b}_{\text{VFM}} \in \mathbb{R}^{D_{\text{VFM}}}$ are the linear layer's trainable parameters.

Encoder features $\mathbf{F}$ are aligned to the features that a VFM produces, $\mathbf{F}_{\text{VFM}}$, when given the same LiDAR image, which encourages the encoder to match representations produced by a semantically strong VFM trained on internet-scale image data. Therefore, the VFM distillation loss is the cosine similarity between $\mathbf{F}$ and $\mathbf{F}_{\text{VFM}}$, following the insight of Wozniak et al. (2026) that the mentioned loss works significantly better for distillation than other loss functions. This is formalized in equation 6,

$$\mathcal{L}_{\text{distill}} = 1 - \frac{1}{1 + H_p W_p} \sum_{k=0}^{H_p W_p} \frac{\langle \mathbf{F}_{k,:}, \mathbf{F}_{\text{VFM},k,:} \rangle}{\|\mathbf{F}_{k,:}\|_2 \, \|\mathbf{F}_{\text{VFM},k,:}\|_2}. \tag{6}$$

MAE helps to learn geometric consistency and forces global context usage. Combined with the distillation task, it allows the model to learn the properties of LiDAR data and mitigates potential artifacts caused by the VFM not having been trained on LiDAR data before. We mask areas of the input, $\mathbf{X}$, randomly, by setting them to zero. Inspired by He et al. (2022), we leave approximately 20% of valid-return pixels visible to the encoder, thus masking such that the masked plus no-return area adds up to 80%. In practice, this implies about 70% true masking, given that approximately 10% of pixels are no-returns on average. The VFM input, $\mathbf{X}_{\text{VFM}}$, is never masked.

We attach a lightweight linear projector layer to our encoder, which spatially upsamples the encoder patch features $\mathbf{Z}^{\text{patch}} \in \mathbb{R}^{(H_p W_p) \times D_{\text{enc}}}$, to the original image size (H, W) via bilinear interpolation

(without trainable parameters),

$$\mathbf{Z}^{\text{patch}\uparrow} = U(\mathbf{Z}^{\text{patch}}), \qquad \mathbf{Z}^{\text{patch}\uparrow} \in \mathbb{R}^{(HW)\times D_{\text{enc}}}, \tag{7}$$

and projects the resulting features from the encoder's embedding space to the image channel space excluding the return channel, $D_{\text{MAE}} = 2$ (for range and intensity), using a minimal, trainable linear projection to encourage encoder learning and prevent offloading learning into the decoder. The reconstructed image $\hat{\mathbf{X}}$ is therefore computed as per equation 8,

$$\hat{\mathbf{X}} = \mathbf{Z}^{\text{patch}\uparrow}\boldsymbol{W}_{\text{MAE}} + \boldsymbol{b}_{\text{MAE}}, \qquad \hat{\mathbf{X}} \in \mathbb{R}^{(HW)\times D_{\text{MAE}}}, \tag{8}$$

where $\boldsymbol{W}_{\text{MAE}} \in \mathbb{R}^{D_{\text{enc}}\times D_{\text{MAE}}}$ and $\boldsymbol{b}_{\text{MAE}} \in \mathbb{R}^{D_{\text{MAE}}}$ are the linear layer's trainable parameters.

We extract the processed range and intensity input channels and flatten them,

$$\mathbf{X}_{\text{rec}} = \text{Flatten}\left(\mathbf{X}_{:,:,\{d,i\}}\right) \in \mathbb{R}^{(HW)\times D_{\text{MAE}}}. \tag{9}$$

and compute the masked autoencoding loss as the sum of pixel-wise mean absolute error and Structural Similarity Index Measure (SSIM) (Wang et al., 2004) components, as per equation 10,

$$\mathcal{L}_{\text{MAE}} = \text{Mask}\left(\left\|\hat{\mathbf{X}} - \mathbf{X}_{\text{rec}}\right\|_1 + \left(1 - \text{SSIM}\left(\hat{\mathbf{X}}, \mathbf{X}_{\text{rec}}\right)\right)\right), \tag{10}$$

Following standard practice (He et al., 2022), the loss is applied only on the masked pixels to evaluate the model's ability to recover masked regions based on global information. Since our masking is intentionally done on valid returns only, we force the model to recover missing information based on available information, with no-returns not being accounted for. This contributes to learning granular semantics and image structure.

Finally, the full loss function reads as in equation 11,

$$\mathcal{L} = \lambda_{\text{distill}}\mathcal{L}_{\text{distill}} + \lambda_{\text{MAE}}\mathcal{L}_{\text{MAE}}, \tag{11}$$

where $\lambda_{\text{distill}} = 1.0$ and $\lambda_{\text{MAE}} = 0.5$ to compensate for the fact that $\mathcal{L}_{\text{distill}}$ contains a single loss term, whereas $\mathcal{L}_{\text{MAE}}$ is the sum of two loss terms. Our SSL goal consists of finding optimal encoder model parameters and auxiliary projection parameters,

$$\hat{\boldsymbol{\theta}}_E, \hat{\boldsymbol{W}}_{\text{VFM}}, \hat{\boldsymbol{W}}_{\text{MAE}}, \hat{\boldsymbol{b}}_{\text{VFM}}, \hat{\boldsymbol{b}}_{\text{MAE}} = \underset{\boldsymbol{\theta}_E, \boldsymbol{W}_{\text{VFM}}, \boldsymbol{W}_{\text{MAE}}, \boldsymbol{b}_{\text{VFM}}, \boldsymbol{b}_{\text{MAE}}}{\arg\min} \mathcal{L}. \tag{12}$$

3.4 Downstream evaluation

Semantic segmentation is a dense task that reflects the quality of the encoder's learned representations. The goal of this task is, for each pixel, to predict to which semantic class among a closed set of classes it belongs. For the sake of evaluation of the representations learned with our SSL approach, we reuse the pre-trained encoder weights, $\hat{\boldsymbol{\theta}}_E$, that generate patch features $\hat{\mathbf{Z}}^{\text{patch}} = E\left(\mathbf{X}; \hat{\boldsymbol{\theta}}_E\right)_{\text{patch}} \in \mathbb{R}^{H_p\times W_p\times D_{\text{enc}}}$, and attach a lightweight semantic segmentation decoder. In some experiments we keep the encoder frozen, and in some we train it further; while the decoder is always trained. The decoder consists of spatial bilinear upsampling (non-trainable) from the patch space, $H_p \times W_p$, into the image space, $H \times W$,

$$\hat{\mathbf{Z}}^{\uparrow} = U_{\text{seg}}(\hat{\mathbf{Z}}^{\text{patch}}), \qquad \hat{\mathbf{Z}}^{\uparrow} \in \mathbb{R}^{H\times W\times D_{\text{enc}}}; \tag{13}$$

a linear projection layer,

$$\mathbf{S} = \hat{\mathbf{Z}}^{\uparrow}\boldsymbol{W}_{\text{seg}} + \boldsymbol{b}_{\text{seg}}, \qquad \mathbf{S} \in \mathbb{R}^{H\times W\times D_{\text{seg}}}, \tag{14}$$

where $\boldsymbol{W}_{\text{seg}} \in \mathbb{R}^{D_{\text{enc}}\times D_{\text{seg}}}$ and $\boldsymbol{b}_{\text{seg}} \in \mathbb{R}^{D_{\text{seg}}}$ are the linear layer's trainable parameters, and D_{seg} equals the number of labeled semantic segmentation classes in the data; and softmax normalization, which produces normalized per-pixel class probabilities,

$$\hat{\mathbf{Y}}_{i,j,c} = \frac{\exp(\mathbf{S}_{i,j,c})}{\sum_{k=1}^{D_{\text{seg}}}\exp(\mathbf{S}_{i,j,k})}, \tag{15}$$

for pixel (i, j) and class c. This layer operates on each patch embedding independently and does not do any mixing between them, to ensure that we evaluate the learned representation rather than decoder strength. Note that the CLS token is not used by this decoder.

The loss is computed against semantic segmentation labels, $\mathbf{Y} \in \{0,1\}^{H \times W \times D_{\text{seg}}}$. Following RangeViT (Ando et al., 2023), we employ the sum of the focal loss and Lovász loss (Berman et al., 2018) applied on the valid labels as a loss function for training,

$$\mathcal{L}_{\text{seg}} = -\frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \sum_{c=1}^{D_{\text{seg}}} \alpha_c (1 - \hat{\mathbf{Y}}_{i,j,c})^{\gamma} \mathbf{Y}_{i,j,c} \log \hat{\mathbf{Y}}_{i,j,c} + \text{Lovasz}\left(\hat{\mathbf{Y}}, \mathbf{Y}\right), \tag{16}$$

with $\gamma = 2$, and α_c the pre-calculated class weight associated with semantic class c such that underrepresented classes are weighted higher, see Section A.3 for details.

In our evaluation, following standard practice, we do Linear Probing (LP), which consists of freezing the encoder and training only the mentioned lightweight decoder using all training labels. We also do scarce-data Fine-Tuning (FT), where both the encoder and decoder are trained, but only a fraction of the training labels are used. Following standard practice, we compute per-class True Positives (TPs), False Positives (FPs), and False Negatives (FNs); and per-class Intersection over Union (IoU) (Everingham et al., 2010) and mean Intersection over Union (mIoU) as evaluation metrics as in equation 17.

$$\text{IoU}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c + \text{FN}_c}, \qquad \text{mIoU} = \frac{1}{D_{\text{seg}}} \sum_{c=1}^{D_{\text{seg}}} \text{IoU}_c. \tag{17}$$

4 Experiments

We perform our experiments on the following datasets and training configurations. The Waymo Open Dataset (Sun et al., 2020) contains 23,691 training and 5,976 validation LiDAR samples annotated with semantic segmentation labels across 22 valid categories. The samples are natively provided as LiDAR images of resolution 64×2650, which during training we downsample to 64×2048 for patch grid-compatibility with other datasets and the VFM. For evaluation, the predictions are upsampled back to the original 64×2650 for fair comparison with the unmodified labels. Although a second return is provided for each sample, for simplicity, we choose to disregard it.

The SemanticKITTI dataset (Behley et al., 2019) contains 19,130 training and 4,071 validation samples annotated with semantic segmentation annotations across 19 valid categories, and 20,351 test samples with held-out labels. By projecting LiDAR points from 3D to the image plane using the SemanticKITTI API (Behley et al., 2019), we create LiDAR images of resolution 64×2048.

Our experiments are run on a single NVIDIA RTX PRO 6000 Blackwell Server Edition GPU with 96 GiB of memory. We use the AdamW optimizer with $\beta = (0.9, 0.999)$ and weight decay 0.01. We pre-train for 30 epochs. The learning rate is scheduled as a warm-up stage starting from 10^{-8} and linearly increasing until reaching $1 \cdot 10^{-3}$ at epoch 5, from which point the learning rate starts to decrease with cosine annealing until reaching 0 at the last epoch. For downstream evaluation, we train for 10 epochs and follow the same learning rate schedule, but peaking at epoch 2. Lastly, for the training data split only, and to mitigate overfitting, we apply horizontal flip with probability 0.5; and random crop with probability 0.5, with a crop scale uniformly sampled within [0.7, 0.9] independently for height and width, resizing back to the pre-crop image resolution via bilinear interpolation for continuous-valued channels and nearest-neighbor interpolation for masks and labels. In trainings where VFM features are used, cropping is done consistently per-patch both on these and on the encoder input. All experiments are initialized with a fixed random seed of 42.

4.1 Downstream semantic segmentation evaluation

We evaluate how the different SSL pre-text tasks and masking strategies contribute to downstream semantic segmentation LP performance in Table 1, and expanded per-class IoU results are in Appendix A.5. The experiments in this table are done initializing the encoder with random weights, pre-training the encoder with our SSL tasks, then freezing the pre-trained encoder and training a linear decoder (LP) on top of it on the full available data.

Table 1: Ablation study on the contribution of pre-text tasks. All results show semantic segmentation mIoU percentage on the Waymo validation set.

Task	Masking	LP
None	None	6.92
Distill	None	**37.65**
	Patch-wise	**37.65**
MAE	Pixel-wise	23.42
	Patch-wise	27.28
Both	Patch-wise	37.52

Table 2: Ablation study on weight initialization of our SSL. All results show semantic segmentation mIoU percentage on the Waymo validation set.

Task	Init. weights	LP
Both	Random	37.52
	ImageNet21k	**38.10**

First, we present the result of initializing the encoder with random weights and doing LP without any pre-training in between as a baseline. This provides a lower bound for LP on untrained features. We then evaluate VFM distillation, both with and without input masking, and find that both score equally, with a gain of over 30 points over the baseline. This enables to combine distillation and MAE seamlessly, as no degradation of downstream performance happens in the distillation task when the input is masked.

As for MAE, since driving scenes often comprise small objects (e.g. pedestrians or traffic signs) far away, we compare masking at the pixel versus patch level, to see if pixel masking helps mitigate complete objects being masked out or mixed up with one another in the auto-encoding task. The answer is negative, as patch masking outperforms pixel masking by more than 3 points, suggesting that pixel masking encourages interpolation rather than global context learning in LiDAR images. Additionally, although MAE shows an improvement of more than 20 points over the baseline, it falls about 10 points short of distillation, showing that the representations learned through MAE are weaker than those distilled from the VFM.

Finally, we combine VFM distillation and MAE with patch masking, and find that the performance is marginally below, but almost equal to, the distillation pre-training alone, showing that MAE does not contribute additional semantic knowledge than the one provided by the VFM distillation.

In Table 2 we evaluate the impact of initializing the encoder with pre-trained weights before our pre-training. Although Ando et al. (2023) showed that initializing RangeViT with image-based pre-trained weights has a significant positive impact for fully supervised, strong-decoder learning; in our SSL, light-decoder regime, we find the impact to be marginal, of just about half a point. Expanded per-class IoU results are in Appendix A.5.

Table 3: Comparison between SSL with LP and partial and full FT against fully supervised training. All results show semantic segmentation mIoU percentage on the Waymo validation set. Best metrics are in **bold** and second-best are underlined.

Method	mIoU	Car	Truck	Bus	Other Veh.	Motorcyclist	Bicyclist	Pedestrian	Sign	Traffic Light	Pole	Constr. Cone	Bicycle	Motorcycle	Building	Vegetation	Tree Trunk	Curb	Road	Lane Marker	Other Ground	Walkable	Sidewalk
Ours (LP)	37.52	82.6	42.1	40.4	10.6	0.1	10.6	45.7	38.0	9.5	28.5	23.9	5.0	5.9	79.6	74.3	33.4	31.6	81.7	32.8	33.2	63.2	52.6
Ours (FT-1%)	37.04	85.3	40.6	19.1	0.0	0.0	0.0	46.0	41.3	10.8	31.4	28.2	0.0	0.0	82.0	76.3	37.5	40.0	84.0	34.2	28.8	69.1	60.4
Ours (FT-5%)	45.77	87.0	46.3	42.0	13.5	0.0	31.0	57.1	50.4	12.9	38.6	27.1	15.8	25.3	85.0	79.2	43.4	50.7	87.2	39.5	34.9	72.3	67.8
Ours (FT-10%)	49.49	88.0	54.1	61.4	**18.8**	0.0	38.7	59.2	52.0	12.6	40.9	31.3	21.1	36.4	85.8	80.3	45.3	53.6	88.0	41.8	36.2	74.0	69.3
Ours (FT-25%)	51.58	**88.5**	**55.7**	67.4	17.8	0.0	45.3	62.8	54.1	13.9	44.1	32.9	23.2	37.7	87.1	81.4	47.9	57.4	89.0	43.9	38.0	75.0	71.6
Ours (FT-50%)	53.49	87.8	55.6	71.3	14.4	0.0	48.5	65.0	56.1	15.4	47.0	36.5	31.3	46.1	87.8	82.3	49.7	59.7	89.6	44.6	39.2	76.0	73.0
RangeViT (IN21k)	55.22	88.2	55.2	**74.5**	17.4	**13.4**	47.0	66.9	57.7	17.2	48.9	39.0	**38.3**	46.2	88.2	82.7	50.6	61.0	89.7	45.2	38.7	76.2	72.7
Ours (FT-100%)	**55.29**	**88.5**	54.5	68.4	18.5	2.5	**51.4**	**67.3**	**58.1**	**17.9**	**49.4**	**40.5**	36.8	**49.3**	**88.6**	**83.2**	**51.5**	**62.4**	**90.2**	**45.9**	**40.7**	**77.0**	**73.8**

Table 3 shows the class-wise and aggregated results for LP and partial and full FT and compared with fully supervised results. All entries in this table use the exact same network architecture and differ only in the learning strategy. The entry RangeViT (IN21k) starts from ImageNet21k's RGB pre-trained ViT weights and does fully supervised training without any SSL. Ours (FT-100%) starts from random initial weights and applies our SSL before fine-tuning using all data. So, our SSL technique

helps RangeViT match and slightly improve representations achieved starting from supervised pre-trained weights. Note that, according to Ando et al. (2023), when initializing the fully supervised training from random instead of ImageNet21k weights, a drop of approximately 2 mIoU points is expected. This means that our SSL method with 50% of FT data is within reach of supervised learning with 100% of data, making our method helpful in scarce-data settings.

Table 4: SemanticKITTI test set semantic segmentation mIoU metrics. We compare our light-decoder, SSL method with full-decoder, fully-supervised prior LiDAR image-based methods.

Method	mIoU	Car	Bicycle	Motorcycle	Truck	Other Veh.	Person	Bicyclist	Motorcyclist	Road	Parking	Sidewalk	Other Ground	Building	Fence	Vegetation	Trunk	Terrain	Pole	Traffic Sign
Ours (LP)	32.6	76.2	8.1	7.0	6.0	14.2	7.8	8.3	1.2	85.4	36.4	58.4	8.8	73.3	35.7	65.9	31.0	52.8	14.6	28.7
Ours (FT 50%)	45.8	83.1	14.9	21.0	13.4	20.7	22.1	28.5	14.5	90.6	57.9	73.2	25.7	83.3	50.8	77.5	50.5	64.7	28.4	50.0
SalsaNext	59.5	91.9	48.3	38.6	38.9	31.9	60.2	59.0	19.4	91.7	63.7	75.8	29.1	90.2	64.2	81.8	63.6	66.5	54.3	62.1
RangeViT	64.0	95.4	55.8	43.5	29.8	42.1	63.9	58.2	38.1	93.1	70.2	80.0	32.5	92.0	69.0	85.3	70.6	71.2	60.8	64.7
RangeFormer	73.3	96.7	69.4	73.7	59.9	66.2	78.1	75.9	58.1	92.4	73.0	78.8	42.4	92.3	70.1	86.6	73.3	72.8	66.4	66.6

Finally, Table 4 compares the performance of our SSL LP with results reported in the literature on the SemanticKITTI test set. As expected, performance is lower than on the Waymo dataset, since SemanticKITTI does not provide LiDAR data natively in image format, which implies back-projection artifacts. Prior art mitigates these artifacts with tailored decoders or 3D clustering post-processing, which our linear decoder without post-processing cannot handle. We attribute the lower performance on the SemanticKITTI dataset to this, and details about the 5-to-10-point mIoU negative impact that can be expected by this can be seen in Appendix A.6. Still, our LP and FT-50% results, the first SSL results on LiDAR range-view images, achieve mIoU scores of 32.6 and 45.8 respectively, which demonstrates that the representation transfers to SemanticKITTI despite a deliberately lightweight decoder, while remaining substantially below specialized supervised segmentation systems.

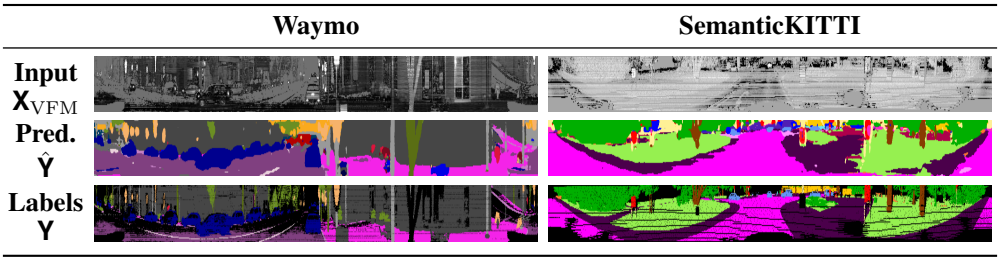

Figure 3: Qualitative semantic segmentation comparison on Waymo and SemanticKITTI.

In Figure 3 we show qualitative results for a validation sample of Waymo and SemanticKITTI. Remarkably, not only does our method produce segmentation predictions that resemble ground truth, but extends plausible predictions into no-return/unlabeled regions, which suggests possible scene completion capabilities attributable to the masking procedures used during ours and the VFM's pretraining. Additional feature and prediction visualizations are provided in Appendix A.1.

5 Conclusion and Future Work

We investigate self-supervised pretraining of LiDAR range-view encoders using DINOv3 feature distillation from LiDAR-derived inputs, optionally combined with MAE. In our experiments, VFM distillation yields substantially higher linear-probe performance than MAE, while the combined objective does not improve over distillation alone; after full fine-tuning, SSL initialization matches the ImageNet-21k-initialized supervised baseline under the reported setup. Thus, LiDAR image-only SSL remains a viable alternative for settings in which sensor modalities other than LiDAR are not available, labels are scarce or absent, or image-based perception architectures are preferred. Future work should evaluate broader geometric and temporal downstream tasks and more advanced distillation or latent-reconstruction objectives.

AI USE STATEMENT

In this work, we used generative AI tools for: design or provide feedback on research methodology or experiments, implement methods, clean and reformat dataset, and interpret results. We have not used generative AI tools for: propose or refine hypotheses, assist with translation, and support qualitative and thematic data analysis; and generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, and assist in the writing of proofs, are not applicable to this work. Additionally, we used generative AI tools for: create or modify scientific figures or images, create or edit software code, creation of artifacts, brainstorming, sourcing/searching for information, edit a research paper to improve readability, and identify relevant literature. We have reviewed all AI-assisted work. We checked LLM-generated research ideas for potential plagiarism through a manual literature survey. LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have described our method in detail. In particular, Sections 3.1 and A.2 describe loading and normalizing the input data, and report exact image resolutions and patch sizes used. Sections 3.2, 3.3, and 3.4 describe which encoder and teacher model are used, both of which are publicly available; as well as auxiliary projection layers and semantic segmentation decoder. These sections walk through each step of the training process; detailing masking strategy, tensor shapes, upsampling operations, trainable parameters, loss functions, and evaluation metrics. Section 4 reports datasets, all of which are publicly available, and data shapes used, as well as training optimizer and hyperparameters used, augmentations applied to training data, and random seeding. After the paper is accepted for publication, to aid seamless reproduction of our results, we will release the code needed to run the experiments.

ACKNOWLEDGMENTS

Acknowledgments will be revealed upon deanonymization after the review process.

REFERENCES

Angelika Ando, Spyros Gidaris, Andrei Bursuc, Gilles Puy, Alexandre Boulch, and Renaud Marlet. Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5240–5250, 2023.

Yeqi Bai, Ben Fei, Youquan Liu, Tao Ma, Yuenan Hou, Botian Shi, and Yikang Li. Rangeperception: Taming lidar range view for efficient and accurate 3d object detection. *Advances in Neural Information Processing Systems*, 36:79557–79569, 2023.

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*, pp. 9297–9307, 2019.

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 *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4413–4421. IEEE, 2018.

Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In *International Symposium on Visual Computing*, pp. 207–222. Springer, 2020.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Ben Fei, Jingyi Xu, Yixuan Li, Weidong Yang, Qingyuan Zhou, Liwen Liu, Tianyue Luo, and Ying He. Self-supervised learning for pre-training 3d point clouds: A survey. *Computational Visual Media*, 2026.

Hariprasath Govindarajan, Maciej K Wozniak, Marvin Klingner, Camille Maurice, B Ravi Kiran, and Senthil Yogamani. Cleverdistiller: Simple and spatially consistent cross-modal distillation. *BMVC*, 2025.

Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9224–9232. Ieee, 2018.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 15979–15988. IEEE, 2022.

Xiang He, Xu Li, Qimin Xu, Longjie Liao, Kun Wei, and Bengwu Wang. Kdrangevit: Knowledge distillation-driven vision transformer for range-view lidar semantic segmentation. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pp. 367–381. Springer, 2025.

International Electrotechnical Commission. Multimedia systems and equipment—colour measurement and management—part 2-1: Colour management—default rgb colour space—srgb, 1999.

Minwoo Jung, Wooseong Yang, Dongjae Lee, Hyeonjae Gil, Giseop Kim, and Ayoung Kim. Helipr: Heterogeneous lidar dataset for inter-lidar place recognition under spatiotemporal variations. *The International Journal of Robotics Research*, 43(12):1867–1883, 2024.

Alexandre Justo Miro, Ludvig af Klinteberg, Bogdan Timus, Aron Asefaw, Ajinkya Khoche, Thomas Gustafsson, Sina Sharif Mansouri, and Masoud Daneshtalab. Correcting and Quantifying Systematic Errors in 3D Box Annotations for Autonomous Driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6724–6732, March 2026.

Alireza G Kashani, Michael J Olsen, Christopher E Parrish, and Nicholas Wilson. A review of lidar radiometric processing: From ad hoc intensity correction to rigorous radiometric calibration. *Sensors*, 15(11):28099–28128, 2015.

Ajinkya Khoche, Gergő László Nagy, Maciej Wozniak, Thomas Gustafsson, and Patric Jensfelt. Blendclip: Bridging synthetic and real domains for zero-shot 3d object classification with multimodal pretraining. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5766–5775. IEEE, 2026.

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*, pp. 228–240, 2023.

Georg Krispel, David Schinagl, Christian Fruhwirth-Reisinger, Horst Possegger, and Horst Bischof. Maeli: Masked autoencoder for large-scale lidar point clouds. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3371–3380. IEEE, 2024.

Paul Julius Kühn, Duc Anh Nguyen, Arjan Kuijper, and Saptarshi Neil Sinha. Rangesam: On the potential of visual foundation models for range-view represented lidar segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1540–1548, 2026.

Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pp. 9–50. Springer, 2002.

Li Li, Khalid N Ismail, Hubert PH Shum, and Toby P Breckon. Durlar: A high-fidelity 128-channel lidar dataset with panoramic ambient and reflectivity imagery for multi-modal autonomous driving applications. In *2021 International Conference on 3D Vision (3DV)*, pp. 1227–1237. IEEE, 2021.

Yingwei Li, Xin Huang, Yang Liu, Yang Fu, Alex Zihao Zhu, Chen Song, Junwen Yao, Anant Subramanian, Hao Xiang, Weijing Shi, et al. Stellar: Scaling 3d perception large models for autonomous driving. *arXiv preprint arXiv:2605.20390*, 2026.

Zhidong Liang, Zehan Zhang, Ming Zhang, Xian Zhao, and Shiliang Pu. Rangeioudet: Range image based real-time 3d object detector optimized by intersection over union. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7140–7149, 2021.

William Ljungbergh, Joakim Johnander, Christoffer Petersson, and Michael Felsberg. Raw or cooked? object detection on raw images. In *Scandinavian Conference on Image Analysis*, pp. 374–385. Springer, 2023.

William Ljungbergh, Adam Lilja, Adam Tonderski, Arvid Laveno Ling, Carl Lindström, Willem Verbeke, Junsheng Fu, Christoffer Petersson, Lars Hammarstrand, and Michael Felsberg. Gasp: Unifying geometric and semantic self-supervised pre-training for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3077–3087, 2026.

Sutharsan Mahendran, Darshana Priyasad, Kaushik Roy, Tharindu Fernando, Sridha Sridharan, Clinton Fookes, and Peyman Moghadam. Tolid: Bridging the architecture gap in vision foundation model to lidar pretraining via token lifting for distillation. *arXiv preprint arXiv:2607.10762*, 2026.

Benedikt Mersch, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. Self-supervised point cloud prediction using 3d spatio-temporal convolutional networks. In *Conference on Robot Learning*, pp. 1444–1454. PMLR, 2022.

Gregory P Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K Wellington. Lasernet: An efficient probabilistic 3d object detector for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12677–12686, 2019.

Gregory P Meyer, Jake Charland, Shreyash Pandey, Ankit Laddha, Shivam Gautam, Carlos Vallespi-Gonzalez, and Carl K Wellington. Laserflow: Efficient and probabilistic object detection and motion forecasting. *IEEE Robotics and Automation Letters*, 6(2):526–533, 2020.

Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 4213–4220. IEEE, 2019.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Ouster Inc. Ouster rev 8 public collection. `https://studio.ouster.com/public/sample_library/public_collection`, 2026. Accessed: 2026-08-31.

Thenukan Pathmanathan, Kanchan Keisham, and Thangarajah Akilan. xmodel-kd: Cross-modal knowledge distillation for 3d scene perception using lidar. *arXiv preprint arXiv:2605.30111*, 2026.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2446–2454, 2020.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Ziyi Wang, Xumin Yu, Yongming Rao, Jie Zhou, and Jiwen Lu. P2p: Tuning pre-trained image models for point cloud analysis with point-to-pixel prompting. *Advances in neural information processing systems*, 35:14388–14402, 2022.

Maciej Wozniak, Jesper Ericsson, Hariprasath Govindarajan, Truls Nyberg, Thomas Gustafsson, Patric Jensfelt, and Olov Andersson. Hilda: Hi erarchical disti l lation with d iffusion for a dvancing self-supervised lidar pre-training. In *European Conference on Computer Vision*, pp. 294–314. Springer, 2026.

Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1887–1893. IEEE, 2018.

Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *2019 international conference on robotics and automation (ICRA)*, pp. 4376–4382. IEEE, 2019.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4840–4851, 2024.

Xiaoyang Wu, Daniel DeTone, Duncan Frost, Tianwei Shen, Chris Xie, Nan Yang, Jakob Engel, Richard Newcombe, Hengshuang Zhao, and Julian Straub. Sonata: Self-supervised learning of reliable point representations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22193–22204, 2025.

Weichen Xu, Jian Cao, Tianhao Fu, Ruilong Ren, Zicong Hu, Xixin Cao, and Xing Zhang. Point cloud reconstruction is insufficient to learn 3d representations. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 8471–8479, 2024a.

Xiang Xu, Lingdong Kong, Hui Shuai, Wenwei Zhang, Liang Pan, Kai Chen, Ziwei Liu, and Qingshan Liu. 4d contrastive superflows are dense 3d representation learners. In *European Conference on Computer Vision*, pp. 58–80. Springer, 2024b.

Honghui Yang, Tong He, Jiaheng Liu, Hua Chen, Boxi Wu, Binbin Lin, Xiaofei He, and Wanli Ouyang. Gd-mae: generative decoder for mae pre-training on lidar point clouds. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9403–9414. IEEE, 2023.

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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19291–19300. IEEE, 2022.

Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21769–21780. IEEE, 2023.

Yujia Zhang, Xiaoyang Wu, Yixing Lao, Chengyao Wang, Zhuotao Tian, Naiyan Wang, and Hengshuang Zhao. Concerto: Joint 2d-3d self-supervised learning emerges spatial representations. *Advances in Neural Information Processing Systems*, 38:69498–69522, 2025.

Zehan Zhang, Zhidong Liang, Ming Zhang, Xian Zhao, Hao Li, Ming Yang, Wenming Tan, and Shiliang Pu. Rangelvdet: Boosting 3d object detection in lidar with range image and rgb image. *IEEE Sensors Journal*, 22(2):1391–1403, 2021.

A Appendix

A.1 Additional visualizations

In Figure 4 we show our encoder's last-stage patch features for the Waymo and Ouster Rev 8 datasets, by mapping Principal Component Analysis (PCA)'s 3 main components to RGB. Specifically, we distill by inputting Ouster Rev 8's RGB channels instead of intensity to the VFM. Although

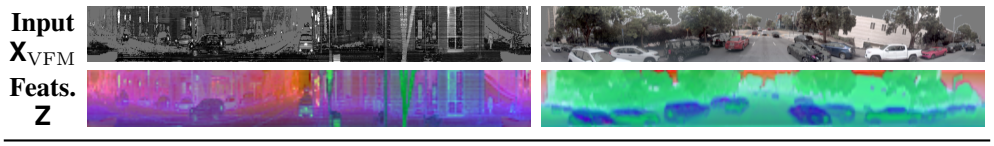

Figure 4: Qualitative comparison of learned representations on Waymo and Ouster Rev 8.

pre-training on Waymo already produces visually good semantic features at the object level, doing so on the higher-resolution, color-rich Ouster Rev 8 data enables our method to produce even crisper features, as even parts of objects such as wheels or doors of cars can be distinguished in the feature map. While Waymo provides a benchmark setting with semantic annotations that allow quantitative evaluation, Ouster Rev 8 illustrates the powerful behavior of our SSL on long-range, high-resolution RGB-LiDAR data.

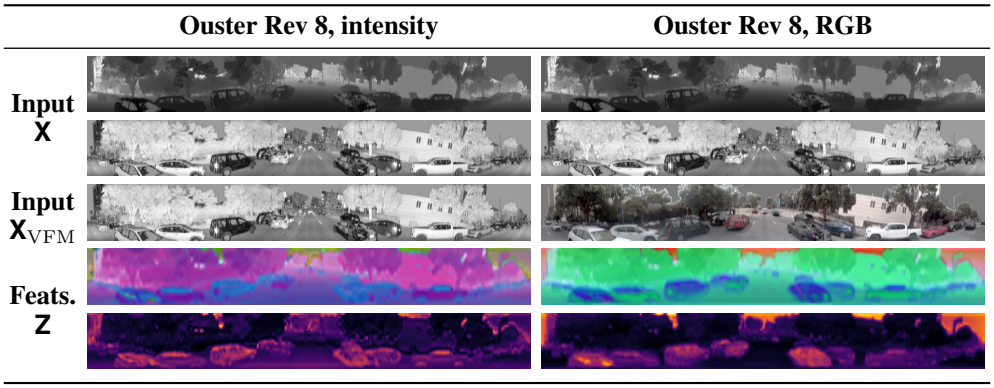

Figure 5: Qualitative comparison of learned representations on Ouster Rev 8, comparing intensity against RGB input for the VFM. From top to bottom, the first two rows are the range and intensity channels that are input to the encoder, the third row is the VFM input, and the last two rows are the patch feature embeddings reduced with PCA and mapped to RGB and their norm, respectively.

Additionally, in Figure 5 we compare patch feature embeddings of our encoder for the same SSL training on Ouster Rev 8 data, just changing the channels that are fed to the VFM during training, comparing intensity with RGB. Although with intensity as input, the features are already of good quality, when the input is RGB we observe that the level of detail is greater, as the features reflect color cues for objects and their parts, which is not seen to that extent when the input is intensity. This shows that very rich representations can be distilled from a VFM into LiDAR range and intensity channels when LiDAR RGB data is available. Thus, there is great potential in perception with modern LiDARs, when even more channels such as RGB itself can be fed to the encoder, where the main limitation at the time of writing is the absence of labels for supervised learning, fine-tuning, and quantitative evaluation.

Finally, in Figure 6 we show the inputs, encoder features, and semantic segmentation predictions and labels for a sample of the Waymo and SemanticKITTI datasets. It can be seen that the encoder features for the SemanticKITTI sample are poorer and capture fewer details than in the Waymo case. We attribute this to Waymo's cleaner, less noisy, better structured input image data, which allows to distill much better representations from the VFM than when the input images are not as clean.

A.2 Input range-view image normalization

For the encoder input, we use the normalization constants of Table 5, which have been precalculated once for each full dataset, based on the processed quantities $\tilde{\mathbf{X}}^d$, $\tilde{\mathbf{X}}^i$. Note that Ouster Rev 8's RGB channels undergo the same post-processing as the intensity, described in Section 3.1.

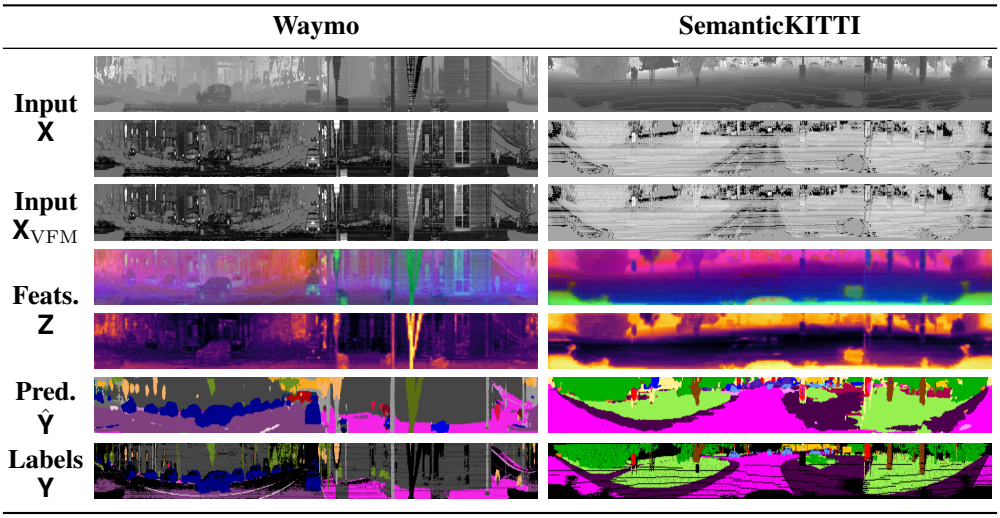

Figure 6: Qualitative comparison of learned representations and downstream semantic segmentation prediction between Waymo and SemanticKITTI. From top to bottom, the first two rows are the range and intensity channels that are input to the encoder, the third row is the VFM intensity input, the next two rows are the patch feature embeddings reduced with PCA and mapped to RGB and their norm, respectively, and the last two rows are the predicted and ground truth semantic segmentation labels, respectively.

Dataset	Channel	Mean, μ_{enc}	Standard deviation, σ_{enc}
Waymo	range	2.8391235581071577	0.6107902006632857
	intensity	0.4394914840973936	0.2231269235096541
SemanticKITTI	range	2.2756053132876266	0.5718762630589120
	intensity	0.6518230476714266	0.2439861186140873
Ouster Rev 8	range	2.8444514668761407	0.7646890757142952
	intensity	0.6189517858283930	0.2181634416275351

Table 5: Normalization constants used in the input image preprocessing described in Section 3.1.

For the VFM input, we use $\mu_{\text{VFM}} = [0.485, 0.456, 0.406]$ and $\sigma_{\text{VFM}} = [0.229, 0.224, 0.225]$ from standard ImageNet normalization for DINOv3's LVD-1689M weights; which the VFM expects (Siméoni et al., 2025).

A.3 Focal loss weights

For the datasets with semantic segmentation labels, we calculate the focal loss weights such that underrepresented classes have a higher weight. In particular, we follow Ando et al. (2023) and compute, for each class c, the frequency in number of points, f_c. We then take the inverse, logarithm, and normalize, as per

$$w_c = \log\left(1 + \frac{1}{f_c}\right). \tag{18}$$

$$\alpha_c = \frac{w_c}{\max w_c} \tag{19}$$

Thus, the α values obtained for Waymo, Table 6; and SemanticKITTI, Table 7; are shown rounded to the second decimal.

A.4 Dataset sequences

We clarify that we used the official training and validation splits of the Waymo Open Dataset.

Class, c	**Car**	**Truck**	**Bus**	**Other Veh.**	**Motorcyclist**	**Bicyclist**	**Pedestrian**	**Sign**	**Traffic Light**	**Pole**	**Constr. Cone**	**Bicycle**	**Motorcycle**	**Building**	**Vegetation**	**Tree Trunk**	**Curb**	**Road**	**Lane Marker**	**Other Ground**	**Walkable**	**Sidewalk**
α_c	0.37	0.66	0.72	0.76	1.00	0.96	0.70	0.72	0.94	0.65	0.95	0.98	0.97	0.21	0.27	0.59	0.63	0.25	0.71	0.73	0.38	0.43

Table 6: Focal loss weights for Waymo Open Dataset semantic segmentation classes.

Class, c	**Car**	**Bicycle**	**Motorcycle**	**Truck**	**Other Veh.**	**Person**	**Bicyclist**	**Motorcyclist**	**Road**	**Parking**	**Sidewalk**	**Other Ground**	**Building**	**Fence**	**Vegetation**	**Trunk**	**Terrain**	**Pole**	**Traffic Sign**
α_c	0.46	0.98	0.96	0.84	0.86	0.96	0.99	1.00	0.26	0.61	0.30	0.77	0.31	0.39	0.23	0.72	0.38	0.81	0.94

Table 7: Focal loss weights for SemanticKITTI semantic segmentation classes.

For the SemanticKITTI dataset, following standard practice, we used sequences 00 to 10 excluding 08 for training, sequence 08 for validation, and sequences 11 to 21 for testing.

Ouster's Rev 8 public collection (Ouster Inc, 2026) contains 15,415 LiDAR samples, which we divide into 14,961 training and 454 validation samples. These are natively provided as LiDAR images of resolution 256×2048 and 256×4096 and, aside from range and intensity, include reflectivity, near-infrared, and RGB channels. No labels are available in this dataset. For extended information about data splits, we refer to Appendix A.4.

For the Ouster Rev 8 public collection, which does not have official splits nor has it been used in prior art for model training, we chose to split the sequences as follows. Training:

- Balmy-alley-sample-1.osf,
- Billboards-RGB_rgb16.osf,
- Chinatown-Busy-Pedestrian-Intersection_rgb16.osf,
- Complex Traffic Signal - Manually managed by Traffic Cops_rgb16.osf,
- OS-1-256-RGB_17baaf33725077dc07fa1c694c0cfcdef97eba9d_2048x10_20260427_142550_rgb16.osf,
- yosemite_valley_rgb16.osf.

Validation:

- Drive-on-Gough-Street-Brakelights-Signal-change.osf

A.5 Full class-wise metrics for ablation studies

Table 8 expands the mIoU results of the main paper with per-class IoU scores. All results are LP using encoder frozen weights according to the SSL method indicated in the first column.

A.6 Full class-wise metrics for SemanticKITTI validation set

In Table 9 we provide SemanticKITTI validation set mIoU and class-wise IoU scores. We provide 2D, range-view evaluation, and 3D evaluation scores. The former is not affected by projection artifacts, as what is found in the image is what becomes evaluated, whereas the latter is degraded due to the fact that when multiple points are projected to the same pixel in the range-view, they do not get an inference label, thus they are regarded as wrong predictions. To mitigate this, prior art has implemented clustering post-processing or tailored decoder architectures, which we leave out of scope. Note that the negative impact of this effect is greater than 5 mIoU points for LP and almost 10 mIoU points for FT 50%.

Table 8: Expanded Waymo validation set semantic segmentation mIoU and IoU for ablation studies.

SSL method	mIoU	Car	Truck	Bus	Other Veh.	Motorcyclist	Bicyclist	Pedestrian	Sign	Traffic Light	Pole	Constr. Cone	Bicycle	Motorcycle	Building	Vegetation	Tree Trunk	Curb	Road	Lane Marker	Other Ground	Walkable	Sidewalk
None	6.92	16.6	17.9	0.0	0.3	0.0	0.0	0.0	3.4	0.0	0.0	0.0	0.0	0.0	31.0	20.7	0.0	0.0	43.4	0.0	0.0	18.1	0.9
Distill	37.65	82.8	45.1	42.8	9.8	0.0	12.1	45.2	38.2	8.8	27.7	24.1	7.2	6.3	80.0	75.3	33.1	28.3	81.1	32.6	32.2	63.8	52.0
Distill masked	37.65	82.8	45.1	42.8	9.8	0.0	12.1	45.2	38.2	8.8	27.7	24.1	7.2	6.3	80.0	75.3	33.1	28.3	81.1	32.6	32.2	63.8	52.0
MAE pixel	23.42	56.6	19.7	5.9	0.9	0.0	1.1	20.4	26.7	5.3	18.8	18.6	3.7	1.0	58.4	56.6	20.1	17.2	69.6	27.5	18.5	42.6	26.0
MAE patch	27.28	67.9	20.9	4.1	1.5	0.0	3.3	24.1	28.9	7.1	20.7	20.9	3.4	3.0	66.0	62.7	23.4	25.6	75.6	30.2	23.2	52.0	35.6
Full	37.52	82.6	42.1	40.4	10.6	0.1	10.6	45.7	38.0	9.5	28.5	23.9	5.0	5.9	79.6	74.3	33.4	31.6	81.7	32.8	33.2	63.2	52.6
Full w/ init. ImageNet21k	38.10	82.9	43.3	43.8	7.5	0.0	12.8	44.8	38.5	10.2	28.9	25.1	8.4	7.3	80.2	75.0	33.5	30.8	81.8	33.0	33.3	63.8	53.2

Table 9: Expanded SemanticKITTI validation set semantic segmentation mIoU and IoU metrics.

Method	Eval.	mIoU	Car	Bicycle	Motorcycle	Truck	Other Veh.	Person	Bicyclist	Motorcyclist	Road	Parking	Sidewalk	Other Ground	Building	Fence	Vegetation	Trunk	Terrain	Pole	Traffic Sign
LP	2D	34.71	83.9	7.5	12.9	11.8	18.9	9.8	14.8	0.0	84.1	25.1	63.9	0.0	69.3	33.0	76.3	35.0	66.6	20.0	26.6
	3D	28.59	70.5	7.2	12.1	10.7	16.2	8.8	13.2	0.0	72.9	22.3	56.6	0.0	48.7	28.3	55.9	26.7	53.8	17.4	21.8
FT 50%	2D	47.02	89.2	9.9	26.6	59.3	26.6	28.5	26.5	0.0	93.4	36.6	79.3	0.2	79.9	49.5	83.9	53.0	75.0	36.8	39.2
	3D	37.94	74.9	9.0	23.2	46.9	20.6	24.3	22.0	0.0	80.7	31.3	69.1	0.1	55.0	41.3	61.3	38.9	60.2	31.3	30.6