Self-Supervised Representation Learning with Joint Embedding Predictive Architecture for Automotive LiDAR Object Detection

Haoran Zhu, Zhenyuan Dong*, Kristi Topollai*, Beiyao Sha*, Anna Ewa Choromanska

[1]Department of Electrical and Computer Engineering, New York University
{hz1922, zd2362, kt2664, bs5125, ac5455}@nyu.edu

Abstract

Recently, self-supervised representation learning relying on vast amounts of unlabeled data has been explored as a pre-training method for autonomous driving. However, directly applying popular contrastive or generative methods to this problem is insufficient and may even lead to negative transfer. In this paper, we present AD-L-JEPA, a novel self-supervised pre-training framework with a joint embedding predictive architecture (JEPA) for automotive LiDAR object detection. Unlike existing methods, AD-L-JEPA is neither generative nor contrastive. Instead of explicitly generating masked regions, our method predicts Bird's-Eye-View embeddings to capture the diverse nature of driving scenes. Furthermore, our approach eliminates the need to manually form contrastive pairs by employing explicit variance regularization to avoid representation collapse. Experimental results demonstrate consistent improvements on the LiDAR 3D object detection downstream task across the KITTI3D, Waymo, and ONCE datasets, while reducing GPU hours by 1.9×–2.7× and GPU memory by 2.8×–4× compared with the state-of-the-art method Occupancy-MAE. Notably, on the largest ONCE dataset, pre-training on 100K frames yields a 1.61 mAP gain, better than in case of all the other methods pre-trained on either 100K or 500K frames, and pre-training on 500K frames yields a 2.98 mAP gain, better than in case of all the other methods pre-trained on either 500K or 1M frames. AD-L-JEPA constitutes the first JEPA-based pre-training method for autonomous driving. It offers better quality, faster, and more GPU-memory-efficient self-supervised representation learning.

Code — https://github.com/HaoranZhuExplorer/adljepa
Extended version — https://arxiv.org/abs/2501.04969

Introduction

Unlike human drivers, current autonomous driving (AD) systems still require large amounts of labeled data for training. This supervised-only paradigm is expensive due to labeling costs and limits the scalability of these systems. Recently, researchers have proposed self-supervised learning (SSL) across camera, LiDAR, and radar modalities (Yang et al. 2024; Min et al. 2023; Zhu et al. 2024) to pre-train the

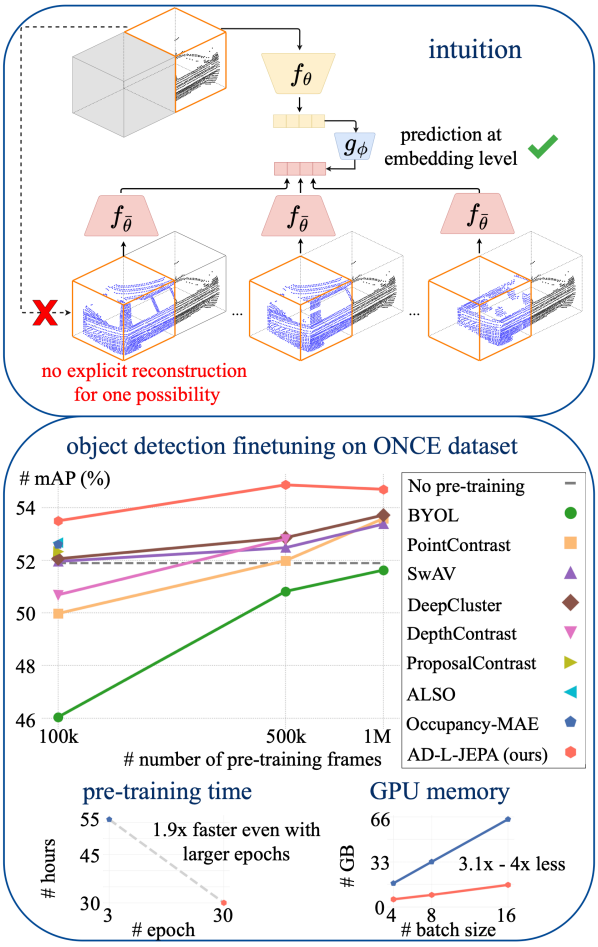

Figure 1: AD-L-JEPA predicts directly in embedding space instead of explicitly reconstructing masked point clouds, as these can correspond to multiple plausible point clouds sharing the same semantics (e.g., "car rear") and can be encoded into the same embedding. It significantly boosts downstream performance while reducing pre-training time and GPU memory usage. The slight saturation of pre-training on 1M frames comes from data redundancies in the ONCE dataset.

*These authors contributed equally.

network without any labels and then fine-tune it with labeled data to adapt to specific downstream tasks. This approach requires less labeled data and improves generalization capability.

In SSL, the two most popular learning paradigms are contrastive methods and generative methods (Balestriero et al. 2023a). However, directly applying these methods for pre-training in AD is challenging and can even hurt downstream performance (Mao et al. 2021). This stems from both the difficulty of defining meaningful contrastive pairs via data augmentation in driving scenarios that contain multiple objects, and the fact that explicit scene generation is time-consuming and insufficient to capture semantic representations of diverse driving scenarios.

In this paper, we present AD-L-JEPA (aka **A**utonomous **D**riving with **L**iDAR data via a **J**oint **E**mbedding **P**redictive **A**rchitecture (LeCun 2022)), a novel self-supervised pre-training framework for automotive LiDAR object detection that, as opposed to existing methods, is neither generative nor contrastive. Our method learns self-supervised representations in Bird's Eye View (BEV) space and predicts embeddings for spatially masked regions. It omits the need to create human-crafted positive/negative pairs, as required by contrastive learning. Furthermore, rather than explicitly reconstructing unknown parts of the data as generative methods do, it predicts BEV embeddings instead.

AD-L-JEPA improves the quality of data representations, capturing high-level semantics and better adapting to uncertainty across multiple plausible driving scenarios (toy illustration capturing that AD-L-JEPA can nicely handle data uncertainty is shown in Figure 1, where different car rears for masked regions are plausible given the visible car front), leading to *better generalization*. It is also extremely *efficient* and *significantly reduces the pre-training time and GPU memory usage*. Extensive experiments show that AD-L-JEPA learns *high-quality embeddings*, which result in *a consistent boost in fine-tuning and transfer learning and offer superior label efficiency for downstream tasks*. A motivating illustration for AD-L-JEPA's is shown in Figure 1.

Our contributions are as follows: AD-L-JEPA introduces the first JEPA-based framework for AD. AD data are scene-centric and sparse. Our novelty lies in adapting JEPA in this setting and employing a BEV-guided masking strategy that enables prediction directly in BEV space while focusing more on non-empty regions. This represents a paradigm shift from 2D image learning to more relevant 3D representations for AD tasks.

Related Work

Self-Supervised Learning

Self-supervised learning (Balestriero et al. 2023b) aims at learning useful data representations from the unlabeled data. We focus here on SSL appraoches that are dedicated to AD with LiDAR point cloud data. Among the existing methods, contrastive techniques learn self-supervised representations by maximizing the feature similarity of manually created positive pairs (matching points) from different levels, such as point level (Xie et al. 2020), depth map level (Zhang et al. 2021), region clusters level (Yin et al. 2022), spatial or temporal segments (Liang et al. 2021; Huang et al. 2021; Nunes et al. 2022; Wu et al. 2023; Nunes et al. 2023; Yuan et al. 2024; Wei et al. 2025; Hegde et al. 2025), or Bird's Eye View (BEV) features over time (Sautier et al. 2024), while minimizing the similarity of negative pairs (non-matched points), if any. Another family of approaches are the generative-based method, where the network is trained to reconstruct masked point clouds (Min et al. 2023; Lin et al. 2024; Xu et al. 2023; Wei et al. 2025; Abdelsamad et al. 2025) or scene surfaces (Boulch et al. 2023; Agro et al. 2024).

Besides contrastive and generative methods, joint-embedding predictive architecture (JEPA) has recently been applied to image and video data modalities (LeCun 2022; Assran et al. 2023; Bardes et al. 2024; Assran et al. 2025) but has not yet been explored for AD scenarios. JEPA learns meaningful representations by predicting the embeddings of the unknown parts of data extracted by a target encoder given the known parts embeddings extracted by a context encoder. Unlike contrastive learning, it does not require complicated pre-processing to create positive pairs and does not suffer from the curse of dimensionality with negative pairs (LeCun 2022). Furthermore, it effectively captures the high uncertainty of the environment without explicitly generating reconstructions of unknown regions that is typically done by generative methods.

Representation collapse is a common phenomenon, when the network fails to learn meaningful representations. It can manifest in two ways: complete collapse, where all representations reduce to a constant vector and thus become useless, or dimensional collapse, where the dimensions of learned representations are highly correlated with each other and contain redundant information. Introducing contrastive learning, or applying regularization techniques (Bardes, Ponce, and LeCun 2022) and moving average updates of the target encoder have been proven useful (Grill et al. 2020). In our paper, we employ both regularization techniques and moving average updates of the target encoder to prevent collapse.

LiDAR-Based 3D Object Detection

We use 3D Object Detection as the downstream task to evaluate the effectiveness of our pre-training method for AD with LiDAR data. For most LiDAR-based 3D object detection downstream algorithms, a sparse 3D convolution encoder (Graham and Van der Maaten 2017) first extracts 3D voxel representations from a given point cloud scene. It then reshapes the image over the height dimension and applies 2D dense convolutions to generate BEV representations. Finally, either a single-stage (Yan, Mao, and Li 2018) or a two-stage detection head (Shi, Wang, and Li 2019; Yin, Zhou, and Krahenbuhl 2021; Shi et al. 2020; Deng et al. 2021) is attached for the LiDAR-based 3D object detection task. Following common settings in the literature, we focus on conducting self-supervised pre-training and evaluate the pre-trained feature quality also using the sparse 3D convolution encoder framework.

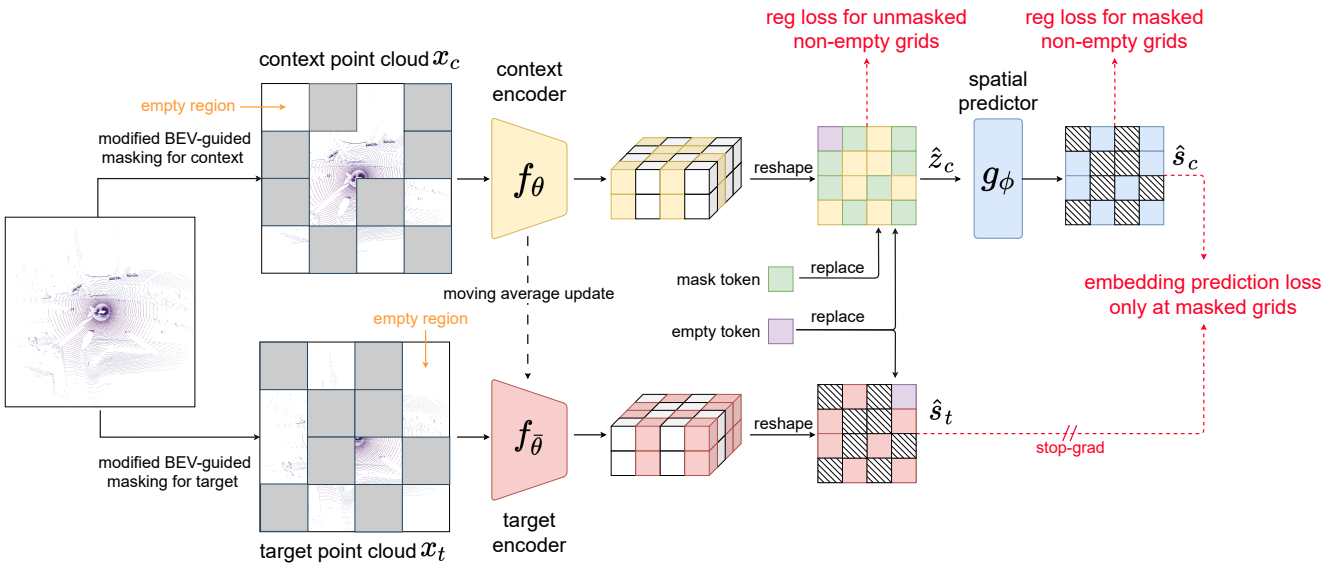

Figure 2: Overview of the AD-L-JEPA architecture: We introduce modified BEV-guided masking to mask the input point cloud in both empty and non-empty regions. The network predicts BEV embeddings at masked regions, leveraging variance regularization at non-empty regions following the output of the context encoder and the lightweight spatial predictor. It also employs a moving average update of the target encoder to learn diverse, high-level semantic representations.

Method

The architecture of AD-L-JEPA is shown in Figure 2. The overarching intuition behind our framework is as follows: for the visible parts of the point cloud scene, the network is trained in a self-supervised manner to predict how the invisible parts should appear in the embedding space. This enables the learning of geometrically and semantically reasonable representations, as well as adapting to the high uncertainty nature of the AD scenes by avoiding the explicit reconstruction of the invisible parts of the data. In the following subsections we describe our design in detail.

Modified BEV-Guided Masking

To learn effective representations in a self-supervised manner, masking is used to create invisible and visible regions. The network is then trained to predict embeddings of the invisible regions based on the visible ones. We have two design recipes for masking in AD scenarios: (1) masks are first created in the BEV embedding space and recursively upsampled to the input point cloud to identify points to be masked; (2) both empty and non-empty areas should be included in the visible and invisible regions created by the masks. These two criteria can be achieved by modifying the BEV-guided masking originally proposed in (Lin et al. 2024) (see comparison in Figure 3).

The original BEV-guided masking involves projecting points onto non-overlapping BEV grids, creating masks only in non-empty BEV grids, and reconstructing only in the masked non-empty regions. This assumes that the network already knows which BEV grid is empty, which stands in contrast with our goal to learn representations to predict all invisible regions. As opposed to this approach, we create

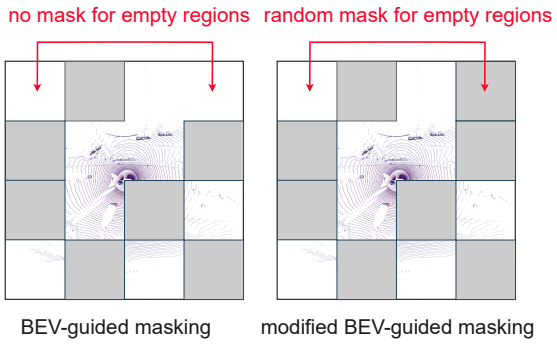

Figure 3: Comparison of original BEV-guided masking (Lin et al. 2024) with our modified version that creates masks in both empty and non-empty regions.

masks for both empty and non-empty BEV grids, resulting in visible and invisible parts that contain both empty and non-empty regions. Consequently, the network is trained to predict the embeddings of all invisible parts, including those that contain empty regions, which enhances the encoder's representation power and helps to learn good quality representations. In our experiments, masks are applied to 50% of non-empty BEV grids and 50% of empty BEV grids. The input points sent to the encoder remain the same as in the original BEV-guided masking. However, the encoder's output BEV embeddings will later be replaced by mask tokens for both masked empty and non-empty grids, whereas in (Lin et al. 2024), only masked non-empty grids were replaced by mask tokens. We denote the unmasked

point cloud as $\boldsymbol{x}_c = \{\boldsymbol{x}_c^1, \ldots, \boldsymbol{x}_c^N\}$ and the masked point cloud as $\boldsymbol{x}_t = \{\boldsymbol{x}_t^1, \ldots, \boldsymbol{x}_t^N\}$ in the multi-batch setting with a batch size of N. $\boldsymbol{x}_c$ and $\boldsymbol{x}_t$ are then sent to the context encoder and target encoder, respectively.

Context Encoder & Target encoder

The context encoder f_θ and target encoder $f_{\bar{\theta}}$ are backbones responsible for extracting context embeddings from the unmasked point cloud and target embeddings from the masked point cloud, respectively. The context encoder will later be used for fine-tuning on the downstream tasks after self-supervised representation learning. It receives input point cloud features and outputs embeddings in a downsampled 3D space $\in \mathbb{R}^{N\times H\times W\times D\times C}$, where H, W, and D represent the length, width, and height dimensions of the 3D embedding, and C is the embedding dimension. We obtain BEV embeddings by reshaping the 3D embeddings. The context BEV embedding is denoted as $\boldsymbol{z}_c = \mathsf{reshape}(f_\theta(\boldsymbol{x}_c)) \in \mathbb{R}^{N\times H\times W\times D\cdot C}$, and the target BEV embeddings as $\boldsymbol{s}_t = \mathsf{reshape}(f_{\bar{\theta}}(\boldsymbol{x}_t)) \in \mathbb{R}^{N\times H\times W\times D\cdot C}$.

Learnable Empty Token and Mask Token

We introduce a learnable empty token ($\in \mathbb{R}^{D\cdot C}$) and a learnable mask token ($\in \mathbb{R}^{D\cdot C}$). By forwarding the unmasked point cloud to the context encoders and the masked point cloud to the target encoders, and after reshaping, we replace all unmasked empty grids in $\boldsymbol{z}_c$ and $\boldsymbol{s}_t$ with the learnable empty token. Simultaneously, we replace all masked grids in $\boldsymbol{z}_c$ with the learnable mask token. We then apply L_2 normalization to each BEV grid's embedding dimension. We denote the context embeddings and target embeddings after such a process by $\hat{\boldsymbol{z}}_c \in \mathbb{R}^{N\times H\times W\times D\cdot C}$ and $\hat{\boldsymbol{s}}_t \in \mathbb{R}^{N\times H\times W\times D\cdot C}$, respectively.

Predictor

The predictor is a lightweight, three-layer convolutional network g_ϕ that predicts target BEV embeddings from visible context BEV embeddings. We denote the predicted embedding, after the L_2 normalization is applied to each BEV grid's embedding dimension, as $\hat{\boldsymbol{s}}_c = g_\phi(\hat{\boldsymbol{z}}_c)$, where $\hat{\boldsymbol{s}}_c$ is in $\mathbb{R}^{N\times H\times W\times D\cdot C}$ and has the same shape as the target BEV embeddings.

Training

We pre-train the network in a self-supervised manner with two losses to ensure we learn high-quality, non-collapsed embeddings: a cosine similarity-based embedding prediction loss and a variance regularization loss.

For predicted embeddings $\hat{\boldsymbol{s}}_c$, we are primarily concerned with the embeddings of masked BEV grids, as we already know the embedding of unmasked BEV grids. Thus, we minimize the cosine similarity-based embedding prediction loss only for the masked grids. We also notice that objects in AD scenarios are distributed in a sparse manner; the number of BEV grids mapped with no points (empty grids) is significantly larger than the number of BEV grids mapped with points (non-empty grids). Therefore, we introduce the hyperparameters $\alpha_0 = 0.25$ and $\alpha_1 = 0.75$ to balance the loss. Overall, the prediction loss is:

$$\begin{aligned}\mathcal{L}_{\text{jepa}} = &\frac{\alpha_0}{\sum_{n=1}^{N}|P_n|}\sum_{n=1}^{N}\sum_{i=1}^{|P_n|}\left(1-\frac{\hat{\boldsymbol{s}}_c^n[i]\cdot\hat{\boldsymbol{s}}_t^n[i]}{\|\hat{\boldsymbol{s}}_c^n[i]\|\,\|\hat{\boldsymbol{s}}_t^n[i]\|}\right)\\ &+\frac{\alpha_1}{\sum_{n=1}^{N}|Q_n|}\sum_{n=1}^{N}\sum_{j=1}^{|Q_n|}\left(1-\frac{\hat{\boldsymbol{s}}_c^n[j]\cdot\hat{\boldsymbol{s}}_t^n[j]}{\|\hat{\boldsymbol{s}}_c^n[j]\|\,\|\hat{\boldsymbol{s}}_t^n[j]\|}\right),\end{aligned} \tag{1}$$

where P_n is the subset containing all indices of masked empty BEV grids, and Q_n is the subset containing all indices of masked non-empty BEV grids in the n-th input in a multi-batch setting. We also denote K_n as the subset containing all indices of unmasked non-empty BEV grids in the n-th input.

In order to avoid representation collapse, we apply an additional variance regularization loss proposed by (Bardes, Ponce, and LeCun 2022):

$$\mathcal{L}_{\text{reg}} = \beta_1\sum_{n=1}^{N} v\left(\hat{\boldsymbol{z}}_c^n[K_n]\right) + \beta_2\sum_{n=1}^{N} v\left(\hat{\boldsymbol{s}}_c^n[Q_n]\right), \tag{2}$$

where the function $v(\cdot)$, which uses an arbitrary input 2D embedding matrix $Y \in \mathbb{R}^{M\times C}$ recording an arbitrary number M of embeddings with a fixed embedding dimension C, is defined in (Bardes, Ponce, and LeCun 2022):

$$v(Y) = \frac{1}{C}\sum_{j=1}^{C}\max(0, \gamma - \sqrt{\text{Var}(Y^j)+\epsilon})). \tag{3}$$

This loss ensures that the average variance across all embedding dimensions, for all unmasked non-empty grids' BEV embeddings after the context encoder $\hat{\boldsymbol{z}}_c^n[K_n] \in \mathbb{R}^{|K_n|\times D\cdot C}$, and all masked non-empty grids' BEV embeddings after the predictor $\hat{\boldsymbol{s}}_c^n[Q_n] \in \mathbb{R}^{|Q_n|\times D\cdot C}$, is larger than some threshold γ. This approach prevents the learning of a meaningless constant embedding. Note that the regularization loss must be computed across each input in the multi-batch setting; otherwise, as we empirically observe, the network tends to learn another meaningless solution consisting of N constant embeddings. These embeddings are significantly distinct from each other and are assigned to all non-empty grids in the n-th input, respectively. In this scenario, although the overall variance exceeds the threshold γ, each BEV embedding in the n-th input remains constant, which is meaningless. Averaging the loss across each input helps avoid this issue.

The overall self-supervised learning loss is:

$$\mathcal{L} = \lambda_{\text{jepa}}\mathcal{L}_{\text{jepa}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}} \tag{4}$$

The parameters of the context encoder, the predictor, as well as the learnable mask token and the learnable empty token, are all updated by gradient descent using Equation 4. Meanwhile, the parameters of the target encoder are updated through a moving average of the context encoder's parameters, $\bar{\theta} \leftarrow \eta\bar{\theta} + (1-\eta)\theta$, to further avoid representation collapse.

Experiments

To evaluate our pre-training method, We utilize the OpenPCDet framework (Team 2020) to pre-process the datasets. We conduct experiments on three datasets of increasing scale: the small-scale KITTI3D (Geiger, Lenz, and Urtasun 2012) (7k frames), the medium-scale Waymo (Sun et al. 2020) (30k frames for the 20% subset and 150k frames for the full set), and the large-scale ONCE (Mao et al. 2021) (100k, 500k, and 1M frames for the small, medium, and large splits). We then fine-tune on a LiDAR 3D object detection task using different network architectures, including SECOND (Yan, Mao, and Li 2018), PV-RCNN (Shi et al. 2020), and CenterPoint (Yin, Zhou, and Krahenbuhl 2021).

We compare models trained from scratch and reproduce the most recent state-of-the-art (SOTA) self-supervised pre-training methods, Occupancy-MAE (Min et al. 2023) and ALSO (Boulch et al. 2023) that are open-sourced and contain no running issues. Specifically, we reproduce Occupancy-MAE using their default settings to obtain the pre-trained weights and fine-tune the downstream object detector on the KITTI3D, Waymo, and ONCE 100k datasets. For ALSO, we directly fine-tune the released pre-trained encoder weights for the downstream object detector on the KITTI3D dataset. But for ALSO's ONCE-100k experiments, we observe that the released pre-trained weights lead to significant negative transfer in three random runs, therefore we compare our results with those reported in their paper. The comparison methods use the same fine-tuning settings as ours. For all fine-tuning experiments, we fine-tune the model through three independent runs in default settings and report the best performance, in the same manner as (Sautier et al. 2024) reports their results. The reproduced results may vary from the original papers. Additionally, for the largest ONCE dataset, we include comparisons against the popular general self-supervised learning algorithms BYOL (Grill et al. 2020), PointContrast (Xie et al. 2020), SwAV (Caron et al. 2020), DeepCluster (Caron et al. 2018), and DepthContrast (Zhang et al. 2021), using results from ONCE's official benchmark website. We also include a comparison to ProposalContrast (Yin et al. 2022), using results reported in (Min et al. 2023).

Due to space constraints, additional details, such as dataset descriptions, hyperparameter settings, training protocols, visualizations, further ablation studies on masking ratios, predictor architecture, learnable empty/mask tokens, and variance regularization, as well as detailed comparisons of pre-training efficiency with other methods, etc., can be found in the supplementary material of the extended version (Zhu et al. 2025).

Pre-training Efficiency

Unlike Occupancy-MAE, which uses computationally expensive dense 3D convolutions to reconstruct invisible regions, AD-L-JEPA employs a joint-embedding predictive architecture at the BEV level and omits those layers, resulting in $2.8\times$–$3.4\times$ lower GPU memory usage and $2.7\times$ fewer GPU hours for pre-training on the 20% and 100% splits of the Waymo dataset, and $3.1\times$–$4\times$ lower GPU memory usage and $1.9\times$ fewer GPU hours for pre-training on the ONCE 100k split.

Downstream Fine-tuning Performance

We first conduct self-supervised pre-training on KITTI3D without any labels and then fine-tune the network on the labeled training data from the same KITTI3D dataset using the widely used SECOND (Yan, Mao, and Li 2018) and PV-RCNN (Shi et al. 2020) methods. As shown quantitatively in Table 1, our self-supervised pre-training approach significantly outperforms the baseline that corresponds to training from scratch and also surpasses other competitor algorithms.

Method	Cars	Ped.	Cycl.	Overall	Diff.
SECOND - R_{40} metric					
No pre-training	**81.99**	52.02	65.07	66.36	
Occupancy-MAE	81.15	50.36	**69.74**	67.08	+0.72
ALSO	81.48	52.50	65.95	66.64	+0.28
AD-L-JEPA (ours)	81.68	**54.15**	67.93	**67.92**	**+1.56**
SECOND - R_{11} metric					
No pre-training	**78.89**	53.78	64.93	65.87	
Occupancy-MAE	78.15	52.10	**69.39**	66.55	+0.68
ALSO	78.37	53.94	66.06	66.12	+0.25
AD-L-JEPA (ours)	78.51	**54.86**	67.94	**67.10**	**+1.23**
PV-RCNN - R_{40} metric					
No pre-training	84.65	56.19	72.19	71.01	
Occupancy-MAE	84.34	57.55	71.33	71.07	+0.06
ALSO	84.64	57.09	**73.72**	71.82	+0.81
AD-L-JEPA (ours)	**85.07**	**59.68**	73.02	**72.59**	**+1.58**
PV-RCNN - R_{11} metric					
No pre-training	83.38	57.01	72.65	71.01	
Occupancy-MAE	83.41	58.68	70.97	71.02	+0.01
ALSO	83.48	57.99	**72.89**	71.45	+0.44
AD-L-JEPA (ours)	**83.74**	**59.91**	72.19	**71.95**	**+0.94**

Table 1: 3D detection results on the KITTI3D validation set, reported with the AP (%) metric. The models are pre-trained with and fine-tuned on the KITTI3D dataset.

Next, in Table 2, we report the results of pre-training with 20% or 100% of the Waymo training data set, and fine-tuning with 20% of the Waymo training data set with Centerpoint (Yin, Zhou, and Krahenbuhl 2021). The overall Average Precision across all classes outperforms the baseline trained from scratch, and our reproduced results using Occupancy-MAE.

Finally, Table 3 reports the results of pre-training on the largest ONCE dataset followed by fine-tuning with SECOND. General self-supervised learning methods yield negligible or even negative gains, indicating the need for a domain specific design. AD-L-JEPA pre-trained on 100k frames outperforms all models trained on 100k or 500k frames and nearly matches the best model trained on 1M frames, and when pre-trained on 500k frames it surpasses every competitor, demonstrating scalability and superior self-supervised learning capability. Interestingly, AD-L-JEPA pre-trained on 1M frames, although significantly better than other methods, falls slightly behind AD-L-JEPA

Method	Veh.	Ped.	Cycl.	Overall	Diff.
Centerpoint - AP metric					
No pre-training	63.28	63.95	66.77	64.67	
Occupancy-MAE, 20%	63.20	64.20	67.20	64.87	+0.20
Occupancy-MAE, 100%	63.53	**64.73**	67.77	65.34	+0.67
AD-L-JEPA (ours), 20%	63.18	64.35	67.68	65.07	+0.40
AD-L-JEPA (ours), 100%	**63.58**	64.58	**68.07**	**65.41**	**+0.74**
Centerpoint - APH metric					
No pre-training	62.77	57.98	65.55	62.10	
Occupancy-MAE, 20%	62.70	58.29	66.00	62.33	+0.23
Occupancy-MAE, 100%	63.04	**58.81**	66.55	62.80	+0.70
AD-L-JEPA (ours), 20%	62.68	58.39	66.48	62.51	+0.41
AD-L-JEPA (ours), 100%	**63.07**	58.64	**66.81**	**62.84**	**+0.74**

Table 2: 3D detection results on the Waymo validation set with LEVEL_2 difficulty defined in (Sun et al. 2020), the AP (%) and APH (%) metric. The models are pre-trained with 20% or 100% of the Waymo dataset, as indicated, and are fine-tuned on the Waymo 20% dataset.

Method	Veh.	Ped.	Cycl.	Overall	Diff.
No pre-training	71.19	26.44	58.04	51.89	
SECOND, pre-trained with 100k frames					
BYOL	68.02	19.50	50.61	46.04	-5.85
PointContrast	71.07	22.52	56.36	49.98	-1.91
SwAV	72.71	25.13	58.05	51.96	+0.07
DeepCluster	73.19	24.00	**58.99**	52.06	+0.17
DepthContrast	71.88	23.57	56.63	50.69	-1.20
ProposalContrast	72.99	25.77	58.23	52.33	+0.44
ALSO	71.73	28.16	58.13	52.68	+0.79
Occupancy-MAE	**73.54**	25.93	58.34	52.60	+0.71
AD-L-JEPA (ours)	73.18	**29.19**	58.14	**53.50**	**+1.61**
SECOND, pre-trained with 500k frames					
BYOL	70.93	25.86	55.63	50.82	-1.07
PointContrast	71.39	27.69	56.88	51.99	+0.10
SwAV	72.51	27.08	57.85	52.48	+0.59
DepthContrast	71.92	29.01	57.51	52.81	+0.92
DeepCluster	71.62	29.33	57.61	52.86	+0.97
AD-L-JEPA (ours)	**73.25**	**31.91**	**59.47**	**54.87**	**+2.98**
SECOND, pre-trained with 1M frames					
BYOL	71.32	25.02	58.56	51.63	-0.26
PointContrast	71.87	28.03	**60.88**	53.59	+1.70
SwAV	72.46	29.84	57.84	53.38	+1.49
DeepCluster	72.89	30.32	57.94	53.72	+1.83
AD-L-JEPA (ours),	**73.01**	**31.94**	59.16	**54.70**	**+2.81**

Table 3: 3D detection results on the ONCE validation set with the AP (%) metric. General self-supervised learning leads to negligle or even negative gains; by contrast, AD-L-JEPA pre-trained with 500k frames significantly outperforms models pre-trained with 1M frames.

pre-trained on 500K frames. This small drop aligns with existing literature showing that increasing the number of unlabeled samples consistently boosts performance but saturates at a point (Goyal et al. 2019). Such saturation can be explained by the data redundancy of highly similar driving scenarios in the 1M frame setting. To validate, we took AD-L-JEPA pretrained on 100K frames and tested it on 16K unseen LiDAR samples from the 500K/1M sets. The 500K set showed a higher average loss (0.44 vs. 0.43), implying richer diversity and stronger fine-tuning transfer. Such saturation could potentially be mitigated by increasing the diversity of pre-training data (Al Kader Hammoud et al. 2024).

Transfer Learning and Label Efficiency

Method	Cars	Ped.	Cycl.	Overall	Diff.
20%, SECOND - R_{40} metric					
No pre-training	79.11	44.36	62.55	62.01	
Occupancy-MAE	79.04	43.85	**63.46**	62.12	+0.11
AD-L-JEPA (ours)	**79.48**	**48.48**	61.92	**63.30**	**+1.29**
20%, SECOND - R_{11} metric					
No pre-training	77.84	45.78	62.46	62.03	
Occupancy-MAE	77.63	45.68	**63.66**	62.32	+0.29
AD-L-JEPA (ours)	**78.11**	**49.71**	62.18	**63.33**	**+1.30**
50%, SECOND - R_{40} metric					
No pre-training	81.05	48.75	62.83	64.21	
Occupancy-MAE	81.20	48.17	64.09	64.49	+0.28
AD-L-JEPA (ours)	**81.55**	**50.13**	**64.12**	**65.27**	**+1.06**
50%, SECOND - R_{11} metric					
No pre-training	78.07	50.53	62.95	63.85	
Occupancy-MAE	78.27	49.67	**63.80**	63.91	+0.06
AD-L-JEPA (ours)	**78.38**	**52.04**	63.53	**64.65**	**+0.80**
100%, SECOND - R_{40} metric					
No pre-training	**81.99**	52.02	65.07	66.36	
Occupancy-MAE	81.65	51.51	66.72	66.63	+0.27
Occupancy-MAE†	81.78	48.92	67.34	66.01	-0.35
AD-L-JEPA (ours)	81.83	52.41	66.73	66.99	+0.63
AD-L-JEPA† (ours)	80.92	**52.45**	**69.76**	**67.71**	**+1.35**
100%, SECOND- R_{11} metric					
No pre-training	**78.89**	53.78	64.93	65.87	
Occupancy-MAE	78.47	52.60	67.23	66.10	+0.23
Occupancy-MAE†	78.61	49.49	66.84	64.98	-0.89
AD-L-JEPA (ours)	78.65	53.62	67.12	66.46	+0.59
AD-L-JEPA† (ours)	77.90	**54.01**	**69.30**	**67.07**	**+1.20**

Table 4: Transfer learning experiment: pre-training on Waymo (20% of the data set) and fine-tuning on KITTI3D. We report 3D detection results on the KITTI3D validation set, with the AP (%) metric. Label efficiency is studied when fine-tuning with 20%, 50%, and 100% of the data. † denotes that the first layer of the model is randomly initialized.

In this section, we report transfer learning from the Waymo 20% split to KITTI3D. Each Waymo point has five attributes, whereas each KITTI3D point has only four. When fine-tuning with an encoder pre-trained directly on the Waymo 20% split, we randomly initialize only the first layer (denoted † at 100% labels) and retain the remaining 3D encoder weights. Alternatively, we drop the fifth 'elongation' feature, re-pretrain on the first four attributes, and thus initialize every layer from pre-trained weights. We also explore fine-tune with 20%, 50%, and 100% of KITTI labels in this setting. Table 4 shows that AD-L-JEPA consistently outperforms baselines across different label efficiencies. Notably,

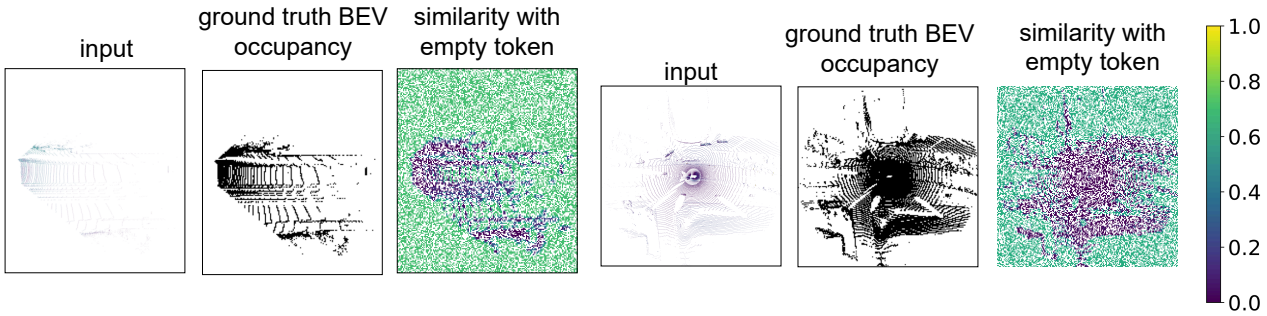

Figure 4: Masked region occupancy estimation evaluated by comparing BEV embeddings obtained by AD-L-JEPA with the learnable empty token via the cosine similarity. Unmasked regions are ignored and the cosine similarity in this case is represented in white color.

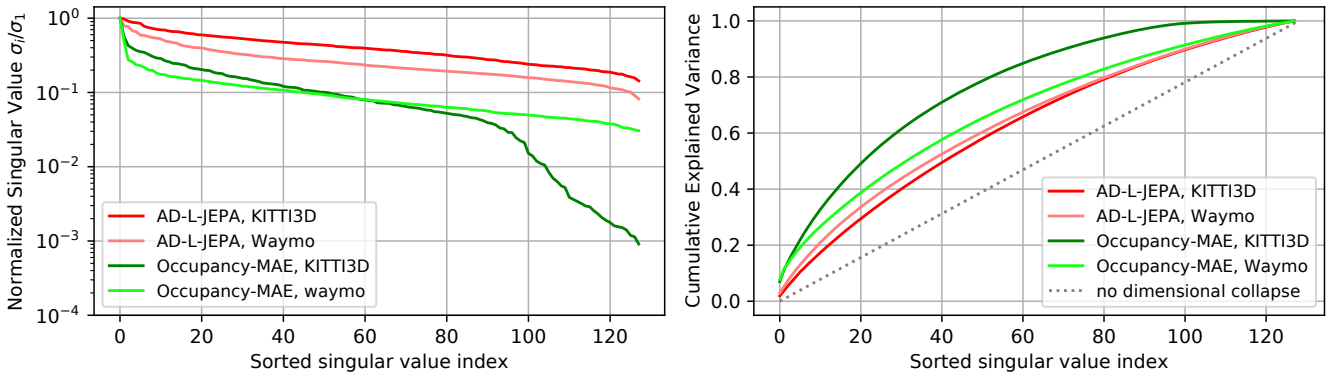

Figure 5: Sorted normalized singular values and the corresponding cumulative explained variance, obtained by singular value decomposition of pre-trained BEV embeddings. Embeddings are obtained either with AD-L-JEPA or Occupancy-MAE.

AD-L-JEPA† model is strongest at 100%, likely because retaining pre-training on all original five features yields richer representations.

Other Evaluations

We also interpret the representation capability of AD-L-JEPA by the following method:

Occupancy Estimation In Figure 4, we estimate the occupancy of masked regions in the input point cloud at a downsampled BEV embedding resolution by comparing each grid's BEV embeddings, outputted by our pre-trained model, against the learnable empty token using cosine similarity. The cosine similarity assigns high similarity to empty grids and low similarity to non-empty grids. We then compare the cosine similarity map with ground truth BEV occupancy side-by-side. Note that the similarity assigned to empty grids by our method is approximately 0.7, not 1, showcasing the model's ability to represent the complex and highly uncertain nature of AD scenes.

Singular Value Decomposition Analysis We follow the methodology described in (Li, Efros, and Pathak 2022) to conduct singular value decomposition of non-empty BEV embeddings outputted by encoders pre-trained with AD-L-JEPA and Occupancy-MAE. This analysis assesses the level of dimensional collapse in pre-trained embeddings. Figure 5 shows that the normalized sorted singular values for AD-L-JEPA are more evenly distributed, while the corresponding cumulative explained variance increases more slowly, indicating less dimensional collapse and less redundant information in the embeddings.

Conclusions

In this paper, we propose AD-L-JEPA, the first joint-embedding predictive architecture for self-supervised representation learning of autonomous-driving LiDAR data. It learns useful representations without any labeled data by predicting masked regions in the BEV embedding space. It neither requires manual creation of positive/negative pairs for contrastive learning nor explicitly reconstructs the complex, high-uncertainty driving scenes. Extensive experiments show that AD-L-JEPA is a more efficient pre-training method for automotive LiDAR-based object detection than state-of-the-art methods, learning richer representations that generalize better to downstream tasks while requiring significantly fewer GPU pre-training hours and less GPU memory. For future work, we plan to extend AD-L-JEPA to leverage temporal dynamics and to incorporate action-conditioned self-supervised representation learning in AD scenarios.

References

Abdelsamad, M.; Ulrich, M.; Gläser, C.; and Valada, A. 2025. Multi-Scale Neighborhood Occupancy Masked Autoencoder for Self-Supervised Learning in LiDAR Point Clouds. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 22234–22243.

Agro, B.; Sykora, Q.; Casas, S.; Gilles, T.; and Urtasun, R. 2024. UnO: Unsupervised Occupancy Fields for Perception and Forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14487–14496.

Al Kader Hammoud, H. A.; Das, T.; Pizzati, F.; Torr, P. H.; Bibi, A.; and Ghanem, B. 2024. On pretraining data diversity for self-supervised learning. In *European Conference on Computer Vision*, 54–71. Springer.

Assran, M.; Bardes, A.; Fan, D.; Garrido, Q.; Howes, R.; Muckley, M.; Rizvi, A.; Roberts, C.; Sinha, K.; Zholus, A.; et al. 2025. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*.

Assran, M.; Duval, Q.; Misra, I.; Bojanowski, P.; Vincent, P.; Rabbat, M.; LeCun, Y.; and Ballas, N. 2023. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15619–15629.

Balestriero, R.; Ibrahim, M.; Sobal, V.; Morcos, A.; Shekhar, S.; Goldstein, T.; Bordes, F.; Bardes, A.; Mialon, G.; Tian, Y.; et al. 2023a. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*.

Balestriero, R.; Ibrahim, M.; Sobal, V.; Morcos, A. S.; Shekhar, S.; Goldstein, T.; Bordes, F.; Bardes, A.; Mialon, G.; Tian, Y.; Schwarzschild, A.; Wilson, A. G.; Geiping, J.; Garrido, Q.; Fernandez, P.; Bar, A.; Pirsiavash, H.; LeCun, Y.; and Goldblum, M. 2023b. A Cookbook of Self-Supervised Learning. *ArXiv*, abs/2304.12210.

Bardes, A.; Garrido, Q.; Ponce, J.; Chen, X.; Rabbat, M.; LeCun, Y.; Assran, M.; and Ballas, N. 2024. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*.

Bardes, A.; Ponce, J.; and LeCun, Y. 2022. VICReg: Variance-Invariance-Covariance Regularization For Self-Supervised Learning. In *ICLR*.

Boulch, A.; Sautier, C.; Michele, B.; Puy, G.; and Marlet, R. 2023. Also: Automotive lidar self-supervision by occupancy estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13455–13465.

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, 132–149.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33: 9912–9924.

Deng, J.; Shi, S.; Li, P.; Zhou, W.; Zhang, Y.; and Li, H. 2021. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI conference on artificial intelligence*.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, 3354–3361. IEEE.

Goyal, P.; Mahajan, D.; Gupta, A.; and Misra, I. 2019. Scaling and benchmarking self-supervised visual representation learning. In *Proceedings of the ieee/cvf International Conference on computer vision*, 6391–6400.

Graham, B.; and Van der Maaten, L. 2017. Submanifold sparse convolutional networks. *arXiv preprint arXiv:1706.01307*.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33: 21271–21284.

Hegde, D.; Lohit, S.; Peng, K.-C.; Jones, M. J.; and Patel, V. M. 2025. Equivariant Spatio-Temporal Self-Supervision for LiDAR Object Detection. In *European Conference on Computer Vision*, 475–491. Springer.

Huang, S.; Xie, Y.; Zhu, S.-C.; and Zhu, Y. 2021. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6535–6545.

LeCun, Y. 2022. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62.

Li, A. C.; Efros, A. A.; and Pathak, D. 2022. Understanding collapse in non-contrastive siamese representation learning. In *European Conference on Computer Vision*, 490–505. Springer.

Liang, H.; Jiang, C.; Feng, D.; Chen, X.; Xu, H.; Liang, X.; Zhang, W.; Li, Z.; and Van Gool, L. 2021. Exploring geometry-aware contrast and clustering harmonization for self-supervised 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3293–3302.

Lin, Z.; Wang, Y.; Qi, S.; Dong, N.; and Yang, M.-H. 2024. BEV-MAE: Bird's Eye View Masked Autoencoders for Point Cloud Pre-training in Autonomous Driving Scenarios. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 3531–3539.

Mao, J.; Niu, M.; Jiang, C.; Liang, H.; Chen, J.; Liang, X.; Li, Y.; Ye, C.; Zhang, W.; Li, Z.; et al. 2021. One million scenes for autonomous driving: Once dataset. *arXiv preprint arXiv:2106.11037*.

Min, C.; Xiao, L.; Zhao, D.; Nie, Y.; and Dai, B. 2023. Occupancy-mae: Self-supervised pre-training large-scale lidar point clouds with masked occupancy autoencoders. *IEEE Transactions on Intelligent Vehicles*.

Nunes, L.; Marcuzzi, R.; Chen, X.; Behley, J.; and Stachniss, C. 2022. SegContrast: 3D point cloud feature representation learning through self-supervised segment discrimination. *IEEE Robotics and Automation Letters*, 7(2): 2116–2123.

Nunes, L.; Wiesmann, L.; Marcuzzi, R.; Chen, X.; Behley, J.; and Stachniss, C. 2023. Temporal consistent 3D lidar representation learning for semantic perception in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5217–5228.

Sautier, C.; Puy, G.; Boulch, A.; Marlet, R.; and Lepetit, V. 2024. Bevcontrast: Self-supervision in bev space for automotive lidar point clouds. In *2024 International Conference on 3D Vision (3DV)*, 559–568. IEEE.

Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10529–10538.

Shi, S.; Wang, X.; and Li, H. 2019. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 770–779.

Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2446–2454.

Team, O. D. 2020. OpenPCDet: An Open-source Toolbox for 3D Object Detection from Point Clouds. https://github.com/open-mmlab/OpenPCDet.

Wei, W.; Nejadasl, F. K.; Gevers, T.; and Oswald, M. R. 2025. T-MAE: temporal masked autoencoders for point cloud representation learning. In *European Conference on Computer Vision*, 178–195. Springer.

Wu, Y.; Zhang, T.; Ke, W.; Süsstrunk, S.; and Salzmann, M. 2023. Spatiotemporal self-supervised learning for point clouds in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5251–5260.

Xie, S.; Gu, J.; Guo, D.; Qi, C. R.; Guibas, L.; and Litany, O. 2020. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, 574–591. Springer.

Xu, R.; Wang, T.; Zhang, W.; Chen, R.; Cao, J.; Pang, J.; and Lin, D. 2023. Mv-jar: Masked voxel jigsaw and reconstruction for lidar-based self-supervised pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13445–13454.

Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10): 3337.

Yang, Z.; Chen, L.; Sun, Y.; and Li, H. 2024. Visual point cloud forecasting enables scalable autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14673–14684.

Yin, J.; Zhou, D.; Zhang, L.; Fang, J.; Xu, C.-Z.; Shen, J.; and Wang, W. 2022. Proposalcontrast: Unsupervised pre-training for lidar-based 3d object detection. In *European conference on computer vision*, 17–33. Springer.

Yin, T.; Zhou, X.; and Krahenbuhl, P. 2021. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11784–11793.

Yuan, J.; Zhang, B.; Yan, X.; Shi, B.; Chen, T.; Li, Y.; and Qiao, Y. 2024. Ad-pt: Autonomous driving pre-training with large-scale point cloud dataset. *Advances in Neural Information Processing Systems*, 36.

Zhang, Z.; Girdhar, R.; Joulin, A.; and Misra, I. 2021. Self-supervised pretraining of 3d features on any point-cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10252–10263.

Zhu, H.; Dong, Z.; Topollai, K.; and Choromanska, A. 2025. AD-L-JEPA: Self-Supervised Spatial World Models with Joint Embedding Predictive Architecture for Autonomous Driving with LiDAR Data. *arXiv preprint arXiv:2501.04969*.

Zhu, H.; He, H.; Choromanska, A.; Ravindran, S.; Shi, B.; and Chen, L. 2024. Multi-View Radar Autoencoder for Self-Supervised Automotive Radar Representation Learning. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, 1601–1608. IEEE.