T-MAE: Temporal Masked Autoencoders for Point Cloud Representation Learning

Weijie Wei, Fatemeh Karimi Nejadasl, Theo Gevers, and Martin R. Oswald

University of Amsterdam, the Netherlands

Abstract. The scarcity of annotated data in LiDAR point cloud understanding hinders effective representation learning. Consequently, scholars have been actively investigating efficacious self-supervised pre-training paradigms. Nevertheless, temporal information, which is inherent in the LiDAR point cloud sequence, is consistently disregarded. To better utilize this property, we propose an effective pre-training strategy, namely Temporal Masked Auto-Encoders (T-MAE), which takes as input temporally adjacent frames and learns temporal dependency. A SiamWCA backbone, containing a Siamese encoder and a windowed cross-attention (WCA) module, is established for the two-frame input. Considering that the movement of an ego-vehicle alters the view of the same instance, temporal modeling also serves as a robust and natural data augmentation, enhancing the comprehension of target objects. SiamWCA is a powerful architecture but heavily relies on annotated data. Our T-MAE pre-training strategy alleviates its demand for annotated data. Comprehensive experiments demonstrate that T-MAE achieves the best performance on both Waymo and ONCE datasets among competitive self-supervised approaches.

Keywords: Self-supervised learning · LiDAR point cloud · 3D detection

1 Introduction

As deep neural networks become more complex, the available amount of labeled data is often insufficient to adequately train huge models [14, 24], *e.g.*, Vision Transformer (ViT) [15]. Consequently, there is a growing interest in exploring self-supervised learning (SSL) approaches as a potential solution to overcome this limitation. SSL serves as a pre-training technique with unlabeled data, accelerating the convergence of the models and improving their performance for downstream tasks [8, 11, 20]. The same challenge extends to the domain of point clouds, where annotations are more costly and time-consuming to obtain [17, 56]. For instance, a mere 10% and 0.8% of frames are annotated in the nuScene [4] and ONCE [35] datasets, respectively. This challenge makes pre-training for point cloud understanding a non-trivial endeavour.

Prior works mainly focus on synthetic and isolated objects [28, 33, 41, 60, 66–68] and indoor scene understanding [25, 30, 34, 43, 59]. Transferring these methods to outdoor LiDAR points is challenging due to their sparsity and dynamic

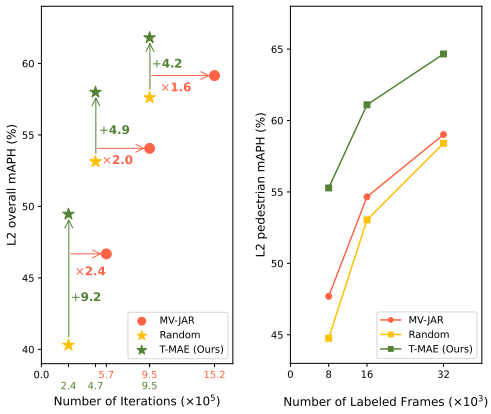


Fig. 1: T-MAE performance on Waymo [47]. Left: Each point triplet shows the performance differences to three models finetuned with the same data. The triplets show finetuned models with 8K, 16K, 32K labeled frames (left to right). Our T-MAE pre-training outperforms both random initialization and the SOTA SSL method MV-JAR [58] with significantly fewer iterations. **Right:** T-MAE yields higher mAPH for pedestrians when finetuned with half the labeled data than MV-JAR.

environmental conditions. At present, most of the self-supervised methods used for understanding point clouds in autonomous driving rely on contrastive learning [38–40, 57, 64]. These approaches model the similarity and dissimilarity between entities, such as segments [39,55] and/or points [57]. In the wake of masked image modeling as a pretext task [24], efforts have also been devoted to the reconstruction of masked points [37,51,58,62]. The main idea is randomly masking points or voxels and urging the network to infer the coordinates of points [62] and/or voxels [58] or other properties, *e.g.*, occupancy [2,37] and curvature [51]. Nevertheless, these methodologies often operate within the confines of a single-frame scenario, disregarding the fact that LiDAR data is typically acquired on a frame-by-frame basis. In other words, the valuable semantic information in temporally adjacent frames is barely exploited.

Several methods attempt to leverage temporal information [27, 31, 39, 55] by incorporating multi-frame input during the self-supervised phase but their core concepts remain grounded in contrastive learning. Specifically, the point clouds captured at different times are treated as augmented samples of the same scene, without including temporal correspondence into the modeling procedure.

Therefore, we propose a new self-supervised paradigm, namely T-MAE, to exploit the accumulated observations. During the pre-training stage, the current scan is voxelized with a high masking ratio, while the previous scan is fed entirely to the encoder. Then, the pretext task is to reconstruct the current scan by incorporating voxel embeddings of the past scan, visible voxel embeddings of the current scan, and the position of masked voxels. This way, the proposed windowed cross-attention module learns to incorporate historical information into the current frame using unlabeled data. The T-MAE pre-training strategy endows the network with both a powerful representation for sparse point clouds and the capacity to strengthen the present by learning from the past. As shown in Fig. 1, the proposed T-MAE achieves higher overall and pedestrian-specific mAPH than the randomly initialized baseline, namely the same model but trained from scratch. Moreover, T-MAE also outperforms state-of-the-art MV-JAR [58] with over 1.6× to 2.4× fewer finetuning iterations.

Our **contributions** are summarized as follows: **1)** We propose T-MAE, a novel and effective SSL approach for representation learning of sparse point clouds, that learns temporal modeling in the process of reconstructing masked points. **2)** We design a SiamWCA backbone, containing a Siamese encoder and a windowed sparse cross-attention (WCA) module, to incorporate historical information. **3)** Our experiments demonstrate the efficacy of T-MAE by attaining substantial improvements on the Waymo and ONCE datasets. Notably, T-MAE with 5% labeled data outperforms the SOTA SSL approach MV-JAR in terms of mAPH for pedestrians, even if MV-JAR employs 10% labeled data.

2 Related Work

Static Self-supervised Learning. SSL for point clouds is a burgeoning field due to the scarcity of annotations. In general, the pipeline of SSL consists of two phases. First, the network is trained using a pretext task with unlabeled data. Second, in downstream tasks such as segmentation and detection, the pretrained weights are loaded to the backbone, and the backbone is attached with task-specific heads. The resulting model is finetuned with annotated data.

The initial research efforts are primarily directed towards contrastive learning. The underlying hypothesis is that an image or 3D scene demonstrates feature equivalence even after undergoing different transformations [7, 11]. One of the key challenges in the point cloud domain is establishing correspondences across scenes. PointContrast [57] and DepthContrast [70] track the points while performing different transformations. GCC-3D [31] exploit sequential information to obtain pseudo instances and then perform contrasting on the instance level. SegContrast [38] firstly obtains segments by an unsupervised clustering and then contrasts segments between two transformed views. These methods allow the network to learn equivalence regarding geometric transformations. However, contrastive SSL approaches usually suffer from careful tuning of hyperparameters and complicated pre- or post-processing to find correspondences.

The focus has shifted to reconstructing masked points, following the success of masked image modeling [24,54]. The masking strategies remain consistent, *i.e.* voxelize a point cloud into cubes followed by random masking of these cubes. The reconstruction targets vary across papers. OccupancyMAE [37] classifies voxels to whether they are occupied. Geo-MAE [51] infers occupancy as well as the normal and curvature of each masked voxel. The GD-MAE's pretext task is to reconstruct a fixed number of points for masked voxels [62]. MV-JAR [58] partitions the masked Voxel into two categories, necessitating the network to either predict the coordinates of the voxels or produce the points themselves. In this paper, we employ the same masking strategy as GD-MAE [62] but expand the knowledge source to include both the current visible voxels and voxels from a previous frame as a reference.

Temporal Self-supervised Learning. There are many SSL methods designed to address temporal or spatiotemporal tasks in the video domain. MAE-ST [18] employs a random patch masking strategy over consecutive frames. The model is

tasked with recovering these masked patches while considering information from adjacent frames. VideoMAE [52] retains the same reconstruction pipeline as the pretext task but exploits a tube masking strategy. SiameseMAE [22] learns object-centred representations with the help of cross-attention layers and an asymmetrical masking technique on consecutive frames. There are also content-based masking strategies, *e.g.*, motion-guided masking [26, 36]. The primary concept underlying these methods is acquiring an understanding of temporal dependency across frames through the process of reconstructing patches with reference to consecutive frames.

While this concept is effectively used in video understanding, it is rarely applied to learn sparse point cloud representations. This is because point clouds are typically handled on a frame-by-frame basis rather than being regarded as a temporal sequence. For instance, when using temporally adjacent scans as input during self-supervised learning, both STSSL [55] and TARL [39] use HDBSCAN [5] to obtain segments and apply self-supervision by minimizing the feature distance between the same segments or points of neighbour frames. While they achieve impressive results, two issues persist 1) The use of elaborate and time-consuming pre-processing methods to obtain segments. 2) A primary focus on object consistency across frames rather than understanding object motion.

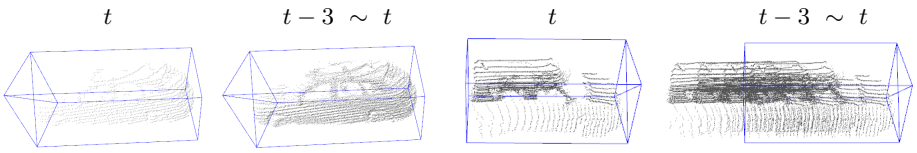
Unlike other approaches, we aim to reduce the reliance on complex pre-processing techniques and focus on enabling the model to establish temporal correspondence through self-supervised learning from unlabeled data.

3 Method

We first briefly discuss important preliminaries in Sec. 3.1 from previous works although they mainly focus on the single-frame setting. To incorporate historical frames, we build up our framework as in Fig. 3. The key component of the framework, SiamWCA, is introduced in 3.2. Then, the proposed T-MAE pre-training strategy is described in Sec. 3.3 and the corresponding windowed sparse cross-attention (WCA) module is elaborated in Sec. 3.4.

3.1 Preliminaries

Pillar-based representation and sparse regional self-attention. The pillar-based representation is an efficient representation introduced in PointPillars [29]. It divides 3D points into infinite-height voxels and computes pillar-wise features. These features can be treated as a pseudo-image in a bird's eye view. On top of the pillar-based representation, SST [16] proposes a sparse regional self-attention (SRA) module to address challenges in applying ViT to sparse LiDAR points. SST divides the pillars into non-overlapping windows and then applies self-attention within each window. The receptive field of each pillar is expanded through a window shift operation. This window partition and shift make the SRA module achieve a good balance of efficiency and accuracy. T-MAE and many recent works [48, 58, 62] are built upon the basic SRA block.



(a) A stationary vehicle. (b) A moving vehicle.

Fig. 2: Comparison between single- and four-frame concatenation. While simple frame concatenation generally improves point density and detection rates, it can introduce spurious points in non-static scene parts that may degrade the detection performance. Since we combine consecutive frames via learned cross-attention, our approach is less affected by this problem. The blue bounding boxes indicate the ground truth for the current frame.

Reconstruction pretext task and single-frame baseline. The original MAE [24] randomly masks patches of an image and employs a ViT to reconstruct the image. To perform this reconstruction pretext task in the LiDAR domain, we follow the concept of the state-of-the-art SSL method GD-MAE [62] which operates on a single frame. The framework encompasses several key stages, *i.e.* voxelization, masking, encoding, dense feature recovery, and reconstruction. Through this pretext task, GD-MAE acquires valuable weights for the encoder, which are later utilized for downstream tasks.

Analysis. Up to this point, a common approach has been to employ a single-frame reconstruction pretext task for LiDAR points. However, incorporating historical frames poses a non-trivial challenge. One straightforward approach is to concatenate two point clouds after aligning them with ego-poses, similar to the three-frame variant of SST [16]. However, as shown in Fig. 2, while concatenation helps to identify static objects effectively, it introduces challenges for detectors when dealing with moving objects. We provide quantitative results in Tab. 3. To address this problem, we suggest learning from past data in a latent space.

3.2 Framework Overview

As illustrated in Fig. 3, the proposed framework compromises voxelization, encoding, windowed cross-attention, dense feature recovery and two separate heads for pre-training and detection, respectively. This subsection elaborates on the key components for supervised learning, *e.g.*, training from scratch and fine-tuning, whereas SSL-related components are introduced in Sec. 3.3.

Temporal batch-based sampling. Consider a sequence of point clouds as an ordered set of point clouds, denoted as $\mathbf{P} = \{\mathcal{P}^1, \mathcal{P}^2, \ldots, \mathcal{P}^t, \ldots, \mathcal{P}^T\}$. In this ordered set, $\mathcal{P}^t = \{(p_k^t)\}_{k=1}^K$ is a sweep of point cloud at time t, where p_k is a 3D point $p_k \in \mathbb{R}^3$ and K indicates the number of points. In our setting, two frames are needed to feed the network. Due to the high sampling frequency of the LiDAR system, two consecutive frames usually contain redundant duplicate information, which is verified in Appendix Sec. S3. Conversely, when there is a substantial time gap between two frames, their overlap may be limited, making the historical

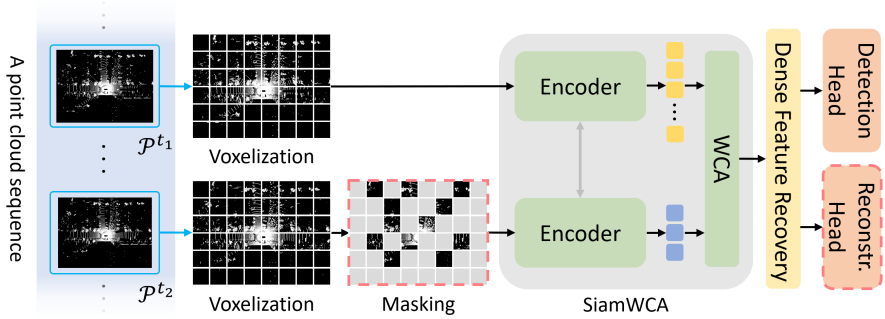


Fig. 3: Overview of our architecture and the proposed T-MAE pre-training. Two frames are sampled from a sequence of point clouds and are voxelized. During pre-training, the current frame $\mathcal{P}^{t_2}$ undergoes an additional masking process. Note that the dashed boxes indicate operations for pre-training phase only. Next, voxel-wise tokens are computed by a Siamese encoder. The two-way gray arrow indicates weight sharing. The WCA module takes as input the full tokens of the previous frame and the partial observation of the current frame and outputs enhanced tokens. The dense feature recovery places sparse tokens back to a dense feature map and convolves the map to fill empty locations. Subsequently, the feature map is either fed to a reconstruction head that recovers masked points, or to a detection head predicting bounding boxes.

information less useful. Therefore, we follow TARL [39] to introduce the concept of a temporal batch so that the interval of two frames is constrained to an appropriate range. Specifically, we sample a batch of consecutive frames $\mathcal{B}^t = \{\mathcal{P}^{t+1}, \mathcal{P}^{t+2}, \ldots, \mathcal{P}^{t+n}\}$ from a sequence $\mathcal{P}$. Then, two frames, $\mathcal{P}^{t_1}$ and $\mathcal{P}^{t_2}$ are sampled from the first and last one-third of the batch, namely, $t_1 \in \{t+1, t+2, \ldots, \lfloor t + \frac{n}{3} \rfloor\}$ and $t_2 \in \{\lceil t + \frac{2n+1}{3} \rceil, \ldots, n-1, n\}$.

Alignment and voxelization. Given two frames, the previous frame $\mathcal{P}^{t_1}$ is transformed to the coordinate system of the current frame $\mathcal{P}^{t_2}$ by their ego-poses. The pose information is available in most datasets [4, 35, 47] or easily obtained by means of GPS/IMU, odometry approaches [9], structure from motion (SfM) algorithms [44, 45], or SLAM systems [13]. Next, each point cloud is divided into discrete voxels. Two linear layers map point coordinates to high-dimensional features and a voxel-wise representation is obtained via voxel-wise average pooling. The voxels can be regarded as pillars, owing to the infinite height.

A Siamese encoder and windowed cross-attention (SiamWCA). A Siamese encoder [3] is a two-branch network where both branches share the same configurations and weights. In this work, it is utilized to encode the pillar-wise representations of both frames to sparse tokens. These tokens serve as input to the WCA module which facilitates the interaction between historical and current tokens. The SiamWCA is elaborated in Sec. 3.4.

Dense feature recovery and detection head. Once the current tokens are augmented with historical information, these sparse tokens are reverted to the

x-y plane to form a dense feature map while vacant pillars are filled with zeros. Since LiDAR points only occur on object surfaces, the object centers typically locate at empty space, leading to inaccurate detection. We follow GD-MAE [62] to attach four dense convolutional layers, which spread the feature from occupied pillars to vacant regions. For the detection head, we adopt a center-based head and use the same target assignment strategy as CenterPoint [65].

3.3 Temporal Masked Autoencoder Pre-training

Up to this point, a two-frame framework for object detection has been set up. However, the transformer-based architecture is data-hungry. Inspired by the SiamMAE [22] used in video understanding, we develop *Temporal Masked Auto-Encoders* (T-MAE) for self-supervised learning on LiDAR points. As shown in Fig. 3, the core idea is to reconstruct the present frame based on a full observation of the historical frame and a partial observation of the current frame. In this way, the network is compelled to learn a powerful sparse representation as well as the capacity to effectively model motion. After the pre-training, the weights of the SiamWCA backbone are retained for the downstream tasks. In the subsequent paragraphs, we elaborate on these steps.

Masking. As shown in Fig. 3, on top of the proposed SiamWCA backbone, an additional step, namely masking, is added between the voxelization and the encoder for the current frame. Specifically, masking is applied in a pillar-wise manner for a high ratio of the occupied pillars, *e.g.*, 75%. The remaining pillars are subsequently input into the encoder. The encoders for the two frames share weights to ensure that the features are constrained in the same latent space. Due to masking, the number of tokens for the current frame decreases significantly, leading to much fewer valid windows and thus accelerating WCA during pre-training. The dense feature recovery performs exactly the same as in Sec. 3.2 and outputs a dense feature map.

Reconstruction head. Given the dense feature map, the reconstruction head retrieves the feature of masked pillars by their spatial location. With the pillar-wise features, the head reconstructs the relative coordinates of points, where the number of output points per pillar is set as K^O. Eventually, the Chamfer distance is computed as the loss function between the reconstructed points and the ground-truth points. Note that the number of points per pillar varies drastically. Thus, a fixed number of points K^{GT} are randomly sampled as the target for reconstruction. In conclusion, T-MAE masks the voxelized tokens of the current frame and compels the network to reconstruct the current frame with the full observation of the historic frame as a reference, which encourages the WCA module to build up correspondence between frames. Note that, unlike the single-frame baseline where only the encoder is reused for downstream tasks, the weights of SiamWCA are fully retained for downstream tasks, which is proved to be effective as shown in Appendix Tab. S10. Therefore, the capability for building up correspondence is also retained.

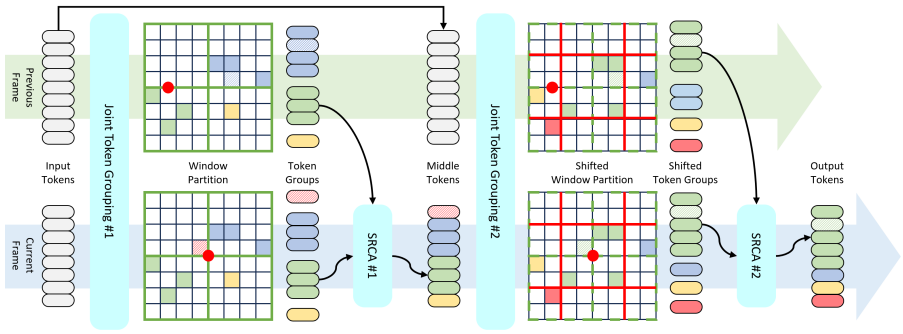


Fig. 4: Windowed sparse cross-attention (WCA). Given the input tokens from both $\mathcal{P}^{t_1}$ and $\mathcal{P}^{t_2}$, a joint token grouping is performed to obtain a window partition. A sparse regional cross-attention (SRCA) is performed independently in each window to integrate the historical information to the middle tokens of the current frame. In other words, the tokens from two frames but with the same colors are attending to each other. For simplicity, the information flow is only depicted for the green tokens. After the second joint token grouping, the cross-attention are performed once more with the shifted window partition. The red dot (●) indicates the ego-vehicle driving towards the right. The box with diagonal stripes (▨) represents an object, *e.g.*, a vehicle, moving towards the left. Best viewed in color and high-resolution.

3.4 Siamese Encoder and Windowed Cross-Attention (SiamWCA)

Siamese encoder. Given the pillar-wise representations of a pair of frames, we explore an asymmetric network and a Siamese encoder for feature encoding. The *asymmetric network* consists of two branches with the same architecture but the branch for $\mathcal{P}^{t_1}$ is modified by reducing the number of channels by half, as depicted in Fig. 6 (a). A *Siamese encoder* is a symmetric network with two sub-networks sharing weights. It is widely used to compute similarity in latent space for tracking [1,21,23,49] and contrastive learning [10,12,42]. For the Siamese encoder, we investigate two weight updating strategies, namely accumulation and SimSiam-style [10]. Accumulation indicates the backward gradients of the two encoders are accumulated. SimSiam, standing for the simple Siamese pre-training strategy [10], indicates the encoder for $\mathcal{P}^{t_1}$ is detached from the computational graph, as depicted in Fig. 6 (b). Thus the weight of this encoder is not updated by gradient propagation but by copying the weights from the encoder for $\mathcal{P}^{t_2}$.

Windowed cross-attention (WCA). When provided with input tokens from two frames, there are several methods for interaction. One intuitive approach is to apply a standard self-attention layer to the concatenation of all tokens from both frames. This approach significantly increases GPU memory requirements because it doubles the number of input tokens. The vanilla Transformer block [53], which consists of a cross-attention layer and a self-attention layer can also be adopted and a single cross-attention layer is feasible as well. However, all these conventional global attentions are not affordable in 3D space due to the expensive computational overhead imposed by the unavoidable high resolution. For

instance, the input for our Siamese encoder can be $\mathbb{R}^{(468\times468)\times128}$, whereas it is $\mathbb{R}^{(14\times14)\times384}$ for a ViT-B/16 due to the patch embedding. Therefore, a window-based implementation is crucial for efficiency. In brief, the WCA module divides the 3D space into non-overlapping windows and then performs cross-attention within the windows. Figure 4 intuitively illustrates this process with a diagram that simulates a scene from a bird's eye view. The key components are elaborated in the following paragraphs.

▷ **Joint token grouping** partitions the 3D space into non-overlapping windows and subsequently allocates the token to the corresponding window based on its spatial location. The previous frame has been aligned with the current frame and thus the window partition is unified for both frames, which allows the following attention mechanism to perform within each window. As shown in Fig. 4, all tokens within the same physical window share the same colour, indicating that they are assigned to the same group for mutual attention.

▷ **Sparse Regional Cross-Attention (SRCA)** is essentially a cross-attention layer where the query comes from $\mathcal{P}^{t_2}$ and the key-value comes from $\mathcal{P}^{t_1}$. For clarity, we consider a single window as an example. Given two groups of tokens $\mathcal{F}^{t_1}, \mathcal{F}^{t_2}$ from two frames and their corresponding spatial coordinates $\mathcal{I}^{t_1}, \mathcal{I}^{t_2}$, the cross-attention is performed as follows:

$$\hat{\mathcal{F}}^{t_2} = \mathbf{MCA}\left(\mathcal{F}^{t_2} + \mathbf{PE}(\mathcal{I}^{t_2}), \mathcal{F}^{t_1} + \mathbf{PE}(\mathcal{I}^{t_1}), \mathcal{F}^{t_1}\right) \tag{1}$$

$$\widetilde{\mathcal{F}}^{t_2} = \mathbf{LN}\left(\mathbf{MLP}\big(\mathbf{LN}(\hat{\mathcal{F}}^{t_2})\big) + \hat{\mathcal{F}}^{t_2}\right) + \mathcal{F}^{t_2} \tag{2}$$

where $\mathbf{MCA}(\mathcal{Q}, \mathcal{K}, \mathcal{V})$ indicates a classical Multi-head Cross-Attention, $\mathbf{PE}(\cdot)$ represents the absolute positional encoding function used in [6], and $\mathbf{LN}(\cdot)$ stands for Layer Normalization. Note that, if a window is empty in the previous frame, the tokens of this window in the current frame will remain unchanged, *i.e.* $\widetilde{\mathcal{F}}^{t_2} = \mathcal{F}^{t_2}$. While the core concept of WCA is identical to cross-attention, this windowed implementation significantly scales down the computational complexity especially when working with sparse pillars.

▷ **Repeated operations with window shift.** Given the middle tokens, the window partition is shifted by half of the window size. Next, the tokens are re-grouped by the joint token grouping, as illustrated in "Shifted Window Partition" of Fig. 4. After that, the SRCA is performed once more. Note that, while performing the SRCA #2, the tokens $\widetilde{\mathcal{F}}^{t_2}$ of $\mathcal{P}^{t_2}$ are updated by SRCA #1 while the tokens $\mathcal{F}^{t_1}$ remain unchanged because a cross-attention operation does not update key $\mathcal{K}$ and value $\mathcal{V}$. With this window shift and SRCA #2, a token interacts with more tokens. For instance, the yellow token in the first window partition only attaches itself in the previous frame via SRCA #1 but interacts with more tokens in SRCA #2.

4 Experiments

4.1 Dataset and Implementation

Experiments are conducted on the following two datasets.

Waymo Open dataset [47] is a large-scale autonomous driving dataset with LiDAR points. For 3D detection, an evaluation protocol is provided for calculating the average precision (AP) and the average precision weighted by heading (APH). Moreover, the evaluation includes two difficulty levels wherein bounding boxes containing over 5 points are regarded as Level 1, while Level 2 indicates all bounding boxes. We adopt mean AP and mean APH at Level 2 as the main evaluation metrics.

ONCE dataset [35] consists of 581 sequences of varying length. 6, 4 and 10 sequences are selected for training, validation and test sets, respectively, and manually annotated. The remaining data is not annotated and adopted for pre-training in our experiments. The official evaluation metrics are AP calculated for each category and range.

Implementation Details. We implement our approach based on the codebase of OpenPCDet[1]. We adopt the sparse pyramid transformer as our encoder and keep the configuration consistent with its official implementation of GD-MAE [62]. Three data augmentation techniques, *i.e.* random flipping, scaling, and rotation, are applied to both frames during pre-training and finetuning. In addition, following MV-JAR [58] and GD-MAE [62], a copy-n-paste augmentation [61] is also employed to slightly address the issue of class imbalance during finetuning. Note that identical data augmentations are applied to a pair of two frames. Further details are in the supplementary material.

4.2 Main Results

We present the comparison with SOTA methods on two datasets: For the ONCE dataset [35], we pre-train our SiamWCA with the *raw_large* split and fine-tune it with the annotated training set. For the Waymo dataset [47], we pre-train SiamWCA with the entire training set. Then, following MV-JAR [58], we finetune our approach with four portions of labeled data, namely 5%, 10%, 20% and 100%. Note that, a strong counterpart, GD-MAE [62], has not been evaluated in this setting and thus we re-train it for a more comprehensive comparison.

The impact of the T-MAE pre-training. The bottom block in Tab. 1 shows that the randomly initialized SiamWCA (denoted as *Random*) performs better than any other models that are initialized with a different pre-training strategy (*i.e.* 69.13 *v.s.* 67.64), suggesting that SiamWCA is a powerful backbone capable of learning temporal modeling when provided with sufficient annotated data. However, its performance drops significantly when finetuning data is limited (*e.g.*, 40.29 *v.s.* 46.68 at 5% level), indicating a strong demand for annotated data. As a comparison, the proposed T-MAE consistently enhances SiamWCA compared to random initialization. Moreover, as the labeled data shrinks from 100% to 5%, the impact of the T-MAE pre-training becomes more pronounced, namely increasing from 1.39 to 9.17, suggesting that the pre-training approach learns a powerful representation and alleviates the demand for annotated data.

[1] `https://github.com/open-mmlab/OpenPCDet`

Table 1: Comparison with SSL methods on the Waymo validation set [47]. Random initialization denotes training from scratch. † represents duplicating the current frame as input during inference. * and ** indicate reproduced by us and taken from [62], respectively. Results for other methods are taken from MV-JAR [58] or the survey [17]. Best results are highlighted as **first**, second, and third. Differences between T-MAE pre-training and random initialization are highlighted in red.

Data Amount	Initialization	Overall		Vehicle		Pedestrian		Cyclist	
		mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH
5%	Random	43.68	40.29	54.05	53.50	53.45	44.76	23.54	22.61
	PointContrast [57]	45.32	41.30	52.12	51.61	53.68	43.22	30.16	29.09
	ProposalContrast [64]	46.62	42.58	52.67	52.19	54.31	43.82	32.87	31.72
	MV-JAR [58]	50.52	46.68	56.47	56.01	57.65	47.69	37.44	36.33
	GD-MAE [62]*	48.23	44.56	56.34	55.76	55.62	46.22	32.72	31.69
	T-MAE†	50.89	47.22	57.06	56.05	58.95	52.62	36.64	32.99
	T-MAE (Ours)	**51.47**$^{\uparrow 7.79}$	**49.46**$^{\uparrow 9.17}$	**57.13**	**56.63**	**59.69**	**55.28**	**37.61**	**36.48**
10%	Random	56.05	53.13	59.78	59.27	60.08	53.04	48.28	47.08
	PointContrast [57]	53.69	49.94	54.76	54.30	59.75	50.12	46.57	45.39
	ProposalContrast [64]	53.89	50.13	55.18	54.71	60.01	50.39	46.48	45.28
	MV-JAR [58]	57.44	54.06	58.43	58.00	63.28	54.66	50.63	49.52
	GD-MAE [62]*	57.67	54.31	59.72	59.19	60.43	52.21	52.85	51.52
	T-MAE†	58.52	55.59	60.26	59.75	62.89	55.85	52.43	51.16
	T-MAE (Ours)	**59.93**$^{\uparrow 3.88}$	**57.99**$^{\uparrow 4.86}$	**60.27**	**59.77**	**65.23**	**61.10**	**54.29**	**53.09**
20%	Random	60.21	57.61	61.58	61.08	64.63	58.41	54.42	53.33
	PointContrast [57]	59.35	55.78	58.64	58.18	64.39	55.43	55.02	53.73
	ProposalContrast [64]	59.52	55.91	58.69	58.22	64.53	55.45	55.36	54.07
	MV-JAR [58]	62.28	59.15	61.88	61.45	66.98	59.02	57.98	57.00
	GD-MAE [62]*	62.32	59.09	62.27	61.79	66.12	58.06	58.57	57.42
	T-MAE†	62.37	60.17	62.19	61.72	67.18	62.18	57.74	56.59
	T-MAE (Ours)	**63.52**$^{\uparrow 3.31}$	**61.80**$^{\uparrow 4.19}$	**63.10**	**62.59**	**68.23**	**64.66**	**59.23**	**58.15**
100%	Random	71.30	69.13	69.05	68.62	73.77	68.80	71.09	69.97
	GCC-3D [31]	65.29	62.79	63.97	63.47	64.23	58.47	67.88	66.44
	BEV-MAE [32]	66.92	64.45	64.78	64.29	66.25	60.53	69.73	68.52
	PointContrast [57]	68.06	64.84	64.24	63.82	71.92	63.81	68.03	66.89
	ProposalContrast [64]	68.17	65.01	64.42	64.00	71.94	63.94	68.16	67.10
	MV-JAR [58]	69.16	66.20	65.52	65.12	72.77	65.28	69.19	68.20
	GD-MAE [62]**	70.62	67.64	68.72	68.29	72.84	65.47	70.30	69.16
	T-MAE†	71.56	69.00	69.39	68.95	74.42	68.43	70.86	69.61
	T-MAE (Ours)	**72.30**$^{\uparrow 1.00}$	**70.52**$^{\uparrow 1.39}$	**69.34**	**68.89**	**75.79**	**72.01**	**71.78**	**70.65**

Comparison with SOTA methods. We aim to leverage the temporal information between two adjacent frames, which is often overlooked by other methods. This absence makes it challenging to compare our method with others in the same setting. Therefore, we implement a test-time single-frame baseline (denoted as T-MAE†) by replicating the same frame and inputting them into our pre-trained model during evaluation. Table 1 shows that T-MAE with identical frames outperforms SOTA counterparts. Moreover, T-MAE with adjacent frames achieves new SOTA at all levels in terms of overall and class-specific metrics. Notably, thanks to the temporal modeling ability, T-MAE significantly outperforms other methods in terms of L2 mAPH for pedestrians. For instance, with 5% labeled data, T-MAE achieve better mAPH (*i.e.* 55.28) than any other method using 10% labeled data. This metric indicates better direction detection for pedestrians, which could benefit downstream applications, *e.g.*, pedestrian intention prediction.

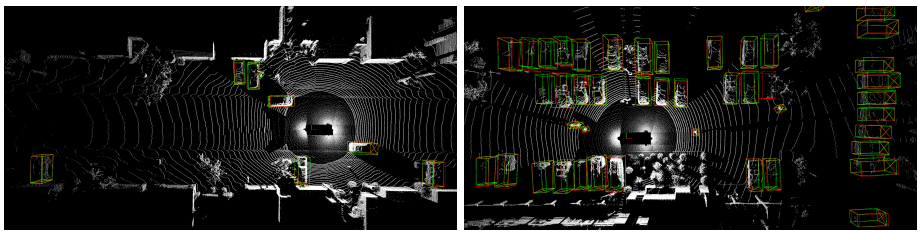


Fig. 5: Qualitative results. We depict ground truth and predictions as boxes colored in red and green for two exemplary scenes from the Waymo dataset [47].

Table 2: Performance comparisons on the validation split of the ONCE dataset [35]. Pt. indicates the model is initialized with pre-trained weights. Results for other methods are taken from GD-MAE [62].

Methods	Pt.	mAP	Vehicle				Pedestrian				Cyclist			
			Overall	0-30m	30-50m	50m-Inf	Overall	0-30m	30-50m	50m-Inf	Overall	0-30m	30-50m	50m-Inf
PV-RCNN [46]	✗	53.55	77.77	**89.39**	72.55	**58.64**	23.50	25.61	22.84	17.27	59.37	71.66	52.58	36.17
IA-SSD [69]	✗	57.43	70.30	83.01	62.84	47.01	39.82	47.45	32.75	18.99	62.17	73.78	56.31	39.53
CenterPoint-Pillar [65]	✗	59.07	74.10	85.23	69.22	53.14	40.94	48.43	34.72	20.09	62.17	73.70	56.05	40.19
CenterPoint-Voxel [65]	✗	60.05	66.79	80.10	59.55	43.39	49.90	56.24	42.61	**26.27**	63.45	74.28	57.94	41.48
SECOND [61]	✗	51.89	71.19	84.04	63.02	47.25	26.44	29.33	24.05	18.05	58.04	69.96	52.43	34.61
w/ BYOL [20]	✓	51.63	71.32	83.59	64.89	50.27	25.02	27.06	22.96	17.04	58.56	70.18	52.74	36.32
w/ PointContrast [57]	✓	53.59$^{\uparrow 1.70}$	71.87	86.93	62.85	48.65	28.03	33.07	25.91	14.44	60.88	71.12	55.77	36.78
w/ DeepCluster [50]	✓	53.72$^{\uparrow 1.83}$	72.89	83.52	67.09	50.38	30.32	34.76	26.43	18.33	57.94	69.18	52.42	34.36
SPT [62]	✗	62.62	75.64	87.21	70.10	53.21	45.92	54.78	37.84	22.56	66.30	78.12	60.52	42.05
w/ GD-MAE [62]	✓	64.92$^{\uparrow 2.30}$	76.79	88.01	71.70	55.60	48.84	58.70	37.30	25.72	69.14	80.29	64.58	45.14
SiamWCA (Ours)	✗	63.71	76.47	87.63	71.59	55.16	47.27	57.57	36.99	21.79	67.40	78.39	62.78	43.90
w/ T-MAE (Ours)	✓	**67.00**$^{\uparrow 3.29}$	**78.35**	88.45	**73.05**	57.16	**52.57**	**62.66**	**44.18**	25.29	**70.09**	**81.14**	**65.33**	**46.48**

To reduce performance variance on 5% and 10% splits, we also report T-MAE performance in the Appendix (see Sec. S4) when it is finetuned with two more 5% and 10% splits and another 20% split. T-MAE consistently outperforms other methods, indicating the efficacy of T-MAE. Since there are no existing multi-frame SSL methods, we also compare our approach with robust non-SSL baselines that utilize multi-frame as inputs in the Appendix (see Tab. S9).

To assess the generalization capabilities of our method, we also conducted experiments on the ONCE dataset [35]. As shown in Tab. 2, T-MAE outperforms other methods in most metrics, indicating its superiority. Moreover, the substantial improvement for pedestrians generalizes to this new dataset, indicating the dominance of T-MAE in pedestrian detection.

Figure 5 shows two exemplary qualitative results from the Waymo dataset and more qualitative results are presented in the Appendix.

4.3 Ablation Study

We perform ablation studies on the Waymo dataset [47] to justify design choices and hyperparameters. For cost-effective experiments, the split 0 with 5% data of the data-efficient benchmark [62] is used to finetune the model by default.

Is a delicate fusion module necessary? To compare with a model taking two frames as input, we modified the input of GD-MAE [62] by concatenating all points of two consecutive frames. We pre-trained, finetuned, and evaluated the modified GD-MAE with the same setting as T-MAE. As depicted in Table 3,

Table 3: Two frames comparison. Two consecutive frames are merged and input into the GD-MAE [62] as an enhanced baseline.

Method	Frame Input		L2 Overall		Vehicle		Pedestrian		Cyclist	
	Previous	Current	mAP	mAPH	AP	APH	AP	APH	AP	APH
GD-MAE	-	0	48.23	44.56	56.34	55.76	55.62	46.22	32.72	31.69
GD-MAE	-	{-1, 0}	47.35	43.69	**57.04**	**56.48**	57.76	48.48	27.26	26.12
T-MAE (Ours)	-1	0	**49.45**	**46.78**	56.56	56.02	**57.96**	**51.94**	**33.82**	**32.37**

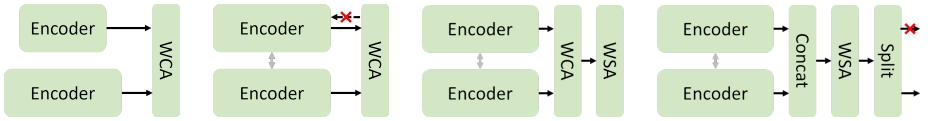
Table 4: Ablation study on model architecture. The proposed SiamWCA (e) demonstrates superior performance in terms of overall mAP and mAPH.

Model	Encoder	Fusion	L2 Overall		Vehicle		Pedestrian		Cyclist	
			mAP	mAPH	AP	APH	AP	APH	AP	APH
(a)	Asymmetric	WCA	47.26	44.78	54.25	53.75	56.66	50.98	30.88	29.60
(b)	SimSiam	WCA	45.58	42.05	53.37	52.84	53.80	44.78	29.58	28.53
(c)	Siamese	WCA+WSA	47.01	45.11	55.81	55.33	**58.95**	**54.69**	26.26	25.31
(d)	Siamese	WSA	43.82	40.90	53.05	52.54	54.08	48.18	24.33	21.97
(e) Ours	Siamese	WCA	**49.45**	**46.78**	**56.56**	**56.02**	57.96	51.94	**33.82**	**32.37**

directly merging two frames improves the metrics in terms of vehicles and pedestrians, which probably results from the density of target objects being doubled, as shown in Fig. 2. However, it has a negative impact on cyclists, resulting in a decrease in overall performance. The drop for cyclists might be attributed to the drift of estimated bounding boxes caused by the fast velocity of bicycles and their relatively small dimensions. In contrast, integrating a historical frame by our method consistently improves overall and class-specific metrics compared to the single-frame baseline.

Backbone design. The proposed architecture consists of a Siamese encoder and a WCA module. We ablate these components with alternatives. **(a)** Asymmetric encoder: it derives from a Siamese encoder but the encoder for $\mathcal{P}^1$ is scaled down. Consequently, the two encoders no longer share weights. **(b)** We employ the SimSiam [10] approach, where one of the two encoders is detached from the computational graph, preventing it from being updated by gradient propagation. The encoder for the current frame is updated by gradients and then shares its weights with the detached encoder. **(c)** The design of the Siamese encoder stays unaltered, while a typical cross-self attention module is adopted in a window-based manner. **(d)** The WCA module in our SiamWCA is replaced with a windowed self-attention (WSA) module. However, since self-attention only needs one input, the tokens of both frames have to be concatenated. Then, the enhanced tokens of the current frame are split from the output of WSA module. These variants are depicted in Fig. 6. The superiority of the proposed SiamWCA is shown in Tab. 4. Variant (c) outperforms SiamWCA in terms of pedestrians but at the expense of both cyclists' performance and extra parameters. Therefore, based on the comprehensive comparison, a Siamese encoder and a WCA fusion module are selected.

Compatibility. To verify that the proposed SSL method T-MAE is independent of a specific encoder or detector, we replace the encoder SPT [62] with two other



(a) Asym. Enc. (b) SimSiam Enc. (c) WCA + WSA (d) WSA

Fig. 6: Four architecture variants of our SiamWCA backbone. (a) Asymmetric encoders: the encoder for the previous frame is scaled down. (b) SimSiam-style [10] encoder: one encoder receives no gradient updates. (c) An additional windows-based self-attention is attached as the classic Transformer [53]. (d) The fusion is implemented by a concat-WSA-split operation.

Table 5: Ablation study on the encoder and the detector. The original encoder SPT [62] is replaced with SST [16] and SpCNN [19] and the original detector enterPoint [65] is replaced with Graph R-CNN [63]. Due to space limits, only overall performance is presented.

Data Amount	Encoder	Detector	Random Init.		T-MAE (Ours)	
			mAP	mAPH	mAP	mAPH
5%	SPT	CenterPoint	43.68	40.29	**51.47**	**49.46**
	SST	CenterPoint	44.26	41.19	**51.59**	**49.24**
	SpCNN	CenterPoint	46.02	43.31	**52.08**	**49.95**
	SPT	Graph R-CNN	51.18	47.27	**56.92**	**54.70**
100%	SPT	CenterPoint	71.30	69.13	**72.30**	**70.52**
	SPT	Graph R-CNN	72.76	70.04	**75.16**	**73.50**

encoders, *i.e.* SST [16] and SpCNN [19], and the detector with a two-stage detector, Graph R-CNN [63]. Table 5 shows that T-MAE constantly improves performance by a significant margin, showcasing its compatibility.

Temporal interval for inference. The temporal interval between two frames should be constrained, as previously discussed in Sec. 3.2. To investigate this matter, we conducted experiments on the Waymo Dataset [47], suggesting that a fixed interval of 0.3 seconds is better than using two consecutive frames. Additional information is available in Appendix Sec. S3.

5 Conclusion

We introduced Temporal Masked Autoencoders (T-MAE), a novel self-supervised paradigm for LiDAR point cloud pre-training. Building upon the single-frame MAE baseline, we incorporated historical frames into the representation using SiamWCA, with the proposed WCA module playing a pivotal role. This pre-training enabled the model to acquire robust representations and the ability to capture motion even with very limited labeled data. By constraining the temporal interval of two frames, we achieved additional performance improvement. Our experiments on the Waymo dataset and the ONCE dataset demonstrate the effectiveness of our approach by showing improvements over state-of-the-art methods.

Acknowledgements

This work was financially supported by TomTom, the University of Amsterdam and the allowance of Top consortia for Knowledge and Innovation (TKIs) from the Netherlands Ministry of Economic Affairs and Climate Policy. This work used the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-7940. Fatemeh Karimi Nejadasl was financed by the University of Amsterdam Data Science Centre.

References

1. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-convolutional siamese networks for object tracking. In: ECCV (2016)
2. Boulch, A., Sautier, C., Michele, B., Puy, G., Marlet, R.: Also: Automotive lidar self-supervision by occupancy estimation. In: CVPR (2023)
3. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a "siamese" time delay neural network. In: NeurIPS. p. 737–744 (1993)
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. In: CVPR (2020)
5. Campello, R.J.G.B., Moulavi, D., Sander, J.: Density-based clustering based on hierarchical density estimates. In: Advances in Knowledge Discovery and Data Mining, vol. 7819, pp. 160–172. Springer Berlin Heidelberg (2013)
6. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV, vol. 12346, pp. 213–229. Springer International Publishing (2020)
7. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: NeurIPS (2021)
8. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: ICCV (2021)
9. Chen, X., Milioto, A., Palazzolo, E., Giguère, P., Behley, J., Stachniss, C.: Suma++: Efficient lidar-based semantic slam. In: IROS. pp. 4530–4537 (2019)
10. Chen, X., He, K.: Exploring simple siamese representation learning. In: CVPR. pp. 15745–15753 (2021)
11. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. In: ICCV (2021)
12. Chen, Z., Huang, G., Li, W., Teng, J., Wang, K., Shao, J., Loy, C.C., Sheng, L.: Siamese detr. In: CVPR (2023)
13. Dellenbach, P., Deschaud, J.E., Jacquet, B., Goulette, F.: Ct-icp: Real-time elastic lidar odometry with loop closure. In: ICRA (2022)
14. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)
15. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
16. Fan, L., Pang, Z., Zhang, T., Wang, Y.X., Zhao, H., Wang, F., Wang, N., Zhang, Z.: Embracing single stride 3d object detector with sparse transformer. In: CVPR (2022)

17. Fei, B., Yang, W., Liu, L., Luo, T., Zhang, R., Li, Y., He, Y.: Self-supervised learning for pre-training 3d point clouds: A survey. arXiv:2305.04691 (2023)
18. Feichtenhofer, C., Fan, H., Li, Y., He, K.: Masked autoencoders as spatiotemporal learners. In: NeurIPS (2022)
19. Graham, B., Engelcke, M., van der Maaten, L.: 3d semantic segmentation with submanifold sparse convolutional networks. CVPR (2018)
20. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M.: Bootstrap your own latent: A new approach to self-supervised learning. In: NeurIPS (2020)
21. Guo, Q., Feng, W., Zhou, C., Huang, R., Wan, L., Wang, S.: Learning dynamic siamese network for visual object tracking. In: ICCV. pp. 1781–1789. IEEE (2017)
22. Gupta, A., Wu, J., Deng, J., Fei-Fei, L.: Siamese masked autoencoders. In: NeurIPS (2023)
23. He, A., Luo, C., Tian, X., Zeng, W.: A twofold siamese network for real-time object tracking. In: CVPR (2018)
24. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: CVPR (2021)
25. Hou, J., Graham, B., Nießner, M., Xie, S.: Exploring data-efficient 3d scene understanding with contrastive scene contexts. In: CVPR (2021)
26. Huang, B., Zhao, Z., Zhang, G., Qiao, Y., Wang, L.: Mgmae: Motion guided masking for video masked autoencoding. In: ICCV (2023)
27. Huang, S., Xie, Y., Zhu, S.C., Zhu, Y.: Spatio-temporal self-supervised representation learning for 3d point clouds. In: ICCV (2021)
28. Jiang, J., Lu, X., Zhao, L., Dazeley, R., Wang, M.: Masked autoencoders in 3D point cloud representation learning. IEEE Transactions on Multimedia (2023)
29. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: PointPillars: Fast encoders for object detection from point clouds. In: CVPR. pp. 12697–12705 (2019)
30. Li, L., Heizmann, M.: A closer look at invariances in self-supervised pre-training for 3d vision. In: ECCV (2022)
31. Liang, H., Jiang, C., Feng, D., Chen, X., Xu, H., Liang, X., Zhang, W., Li, Z., Van Gool, L.: Exploring geometry-aware contrast and clustering harmonization for self-supervised 3d object detection. In: ICCV. pp. 3273–3282 (2021)
32. Lin, Z., Wang, Y.: BEV-MAE: Bird's eye view masked autoencoders for outdoor point cloud pre-training. In: AAAI (2024)
33. Liu, H., Cai, M., Lee, Y.J.: Masked discrimination for self-supervised learning on point clouds. In: ECCV (2022)
34. Liu, L., Zhuang, Z., Huang, S., Xiao, X., Xiang, T., Chen, C., Wang, J., Tan, M.: CPCM: Contextual point cloud modeling for weakly-supervised point cloud semantic segmentation. In: ICCV (2023)
35. Mao, J., Niu, M., Jiang, C., Liang, H., Chen, J., Liang, X., Li, Y., Ye, C., Zhang, W., Li, Z., Yu, J., Xu, H., Xu, C.: One million scenes for autonomous driving: Once dataset. In: NeurIPS (2021)
36. Mao, Y., Deng, J., Zhou, W., Fang, Y., Ouyang, W., Li, H.: Masked motion predictors are strong 3d action representation learners. In: ICCV (2023)
37. Min, C., Xu, X., Zhao, D., Xiao, L., Nie, Y., Dai, B.: Occupancy-MAE: Self-supervised pre-training large-scale lidar point clouds with masked occupancy autoencoders. IEEE Transaction on Intelligent Vehicles (2022)

38. Nunes, L., Marcuzzi, R., Chen, X., Behley, J., Stachniss, C.: SegContrast: 3D point cloud feature representation learning through self-supervised segment discrimination. IEEE Robotics and Automation Letters (RA-L) **7**(2), 2116–2123 (2022)
39. Nunes, L., Wiesmann, L., Marcuzzi, R., Chen, X., Behley, J., Stachniss, C.: Temporal consistent 3d LiDAR representation learning for semantic perception in autonomous driving. In: CVPR (2023)
40. Pang, B., Xia, H., Lu, C.: Unsupervised 3d point cloud representation learning by triangle constrained contrast for autonomous driving. In: CVPR (2023)
41. Pang, Y., Wang, W., Tay, F.E.H., Liu, W., Tian, Y., Yuan, L.: Masked autoencoders for point cloud self-supervised learning. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV. vol. 13662, pp. 604–621. Cham (2022)
42. Peng, X., Wang, K., Zhu, Z., Wang, M., You, Y.: Crafting better contrastive views for siamese representation learning. In: CVPR. pp. 16010–16019. IEEE (2022)
43. Rao, Y., Liu, B., Wei, Y., Lu, J., Hsieh, C.J., Zhou, J.: Randomrooms: Unsupervised pre-training from synthetic shapes and randomized layouts for 3d object detection. In: ICCV (2021)
44. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
45. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: ECCV (2016)
46. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: CVPR (2020)
47. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D.: Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR. pp. 2443–2451 (2020)
48. Sun, P., Tan, M., Wang, W., Liu, C., Xia, F., Leng, Z., Anguelov, D.: Swformer: Sparse window transformer for 3d object detection in point clouds. In: ECCV (2022)
49. Tao, R., Gavves, E., Smeulders, A.W.M.: Siamese instance search for tracking. In: CVPR (2016)
50. Tian, K., Zhou, S., Guan, J.: Deepcluster: A general clustering framework based on deep learning. In: Machine Learning and Knowledge Discovery in Databases. pp. 809–825 (2017)
51. Tian, X., Ran, H., Wang, Y., Zhao, H.: GeoMAE: Masked geometric target prediction for self-supervised point cloud pre-training. In: CVPR (2023)
52. Tong, Z., Song, Y., Wang, J., Wang, L.: VideoMAE: masked autoencoders are data-efficient learners for self-supervised video pre-training. In: NeurIPS (2022)
53. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS p. 11 (2017)
54. Wei, C., Fan, H., Xie, S., Wu, C.Y., Yuille, A., Feichtenhofer, C.: Masked feature prediction for self-supervised visual pre-training. In: CVPR. pp. 14648–14658. IEEE, New Orleans, LA, USA (2022)
55. Wu, Y., Zhang, T., Ke, W., Susstrunk, S., Salzmann, M.: Spatiotemporal self-supervised learning for point clouds in the wild. In: CVPR. pp. 5251–5260 (2023)
56. Xiao, A., Huang, J., Guan, D., Zhang, X., Lu, S., Shao, L.: Unsupervised point cloud representation learning with deep neural networks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(9), 11321–11339 (2023)

57. Xie, S., Gu, J., Guo, D., Qi, C.R., Guibas, L.J., Litany, O.: Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In: ECCV (2020)
58. Xu, R., Wang, T., Zhang, W., Chen, R., Cao, J., Pang, J., Lin, D.: MV-JAR: Masked voxel jigsaw and reconstruction for LiDAR-based self-supervised pre-training. In: CVPR (2023)
59. Yamada, R., Kataoka, H., Chiba, N., Domae, Y., Ogata, T.: Point cloud pre-training with natural 3d structures. In: CVPR. pp. 21251–21261 (2022)
60. Yan, X., Zhan, H., Zheng, C., Gao, J., Zhang, R., Cui, S., Li, Z.: Let images give you more:point cloud cross-modal training for shape analysis. In: NeurIPS (2022)
61. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
62. Yang, H., He, T., Liu, J., Chen, H., Wu, B., Lin, B., He, X., Ouyang, W.: GD-MAE: Generative decoder for MAE pre-training on LiDAR point clouds. In: CVPR (2023)
63. Yang, H., Liu, Z., Wu, X., Wang, W., Qian, W., He, X., Cai, D.: Graph r-cnn: Towards accurate 3d object detection with semantic-decorated local graph. In: ECCV (2022)
64. Yin, J., Zhou, D., Zhang, L., Fang, J., Xu, C.Z., Shen, J., Wang, W.: Proposalcontrast: Unsupervised pre-training for lidar-based 3d object detection. In: ECCV (2022)
65. Yin, T., Zhou, X., Krähenbühl, P.: Center-based 3d object detection and tracking. In: CVPR (2021)
66. Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., Lu, J.: Point-BERT: Pre-training 3d point cloud transformers with masked point modeling. In: CVPR (2022)
67. Zhang, R., Guo, Z., Fang, R., Zhao, B., Wang, D., Qiao, Y., Li, H., Gao, P.: Point-M2AE: Multi-scale masked autoencoders for hierarchical point cloud pre-training. In: NeurIPS (2022)
68. Zhang, R., Wang, L., Qiao, Y., Gao, P., Li, H.: Learning 3D representations from 2D pre-trained models via image-to-point masked autoencoders. In: CVPR (2023)
69. Zhang, Y., Hu, Q., Xu, G., Ma, Y., Wan, J., Guo, Y.: Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In: CVPR (2022)
70. Zhang, Z., Girdhar, R., Joulin, A., Misra, I.: Self-supervised pretraining of 3d features on any point-cloud. In: ICCV (2021)