BEV-MAE: Bird's Eye View Masked Autoencoders for Point Cloud Pre-training in Autonomous Driving Scenarios

Zhiwei Lin[1], Yongtao Wang[1]*, Shengxiang Qi[2], Nan Dong[2], Ming-Hsuan Yang[3]
[1]Wangxuan Institute of Computer Technology, Peking University
[2]Chongqing Changan Automobile Co., Ltd
[3]University of California, Merced
{zwlin, wyt}@pku.edu.cn, shengxiang.qi@gmail.com, dongnan1@changan.com.cn, mhyang@ucmerced.edu

Abstract

Existing LiDAR-based 3D object detection methods for autonomous driving scenarios mainly adopt the training-from-scratch paradigm. Unfortunately, this paradigm heavily relies on large-scale labeled data, whose collection can be expensive and time-consuming. Self-supervised pre-training is an effective and desirable way to alleviate this dependence on extensive annotated data. In this work, we present BEV-MAE, an efficient masked autoencoder pre-training framework for LiDAR-based 3D object detection in autonomous driving. Specifically, we propose a bird's eye view (BEV) guided masking strategy to guide the 3D encoder learning feature representation in a BEV perspective and avoid complex decoder design during pre-training. Furthermore, we introduce a learnable point token to maintain a consistent receptive field size of the 3D encoder with fine-tuning for masked point cloud inputs. Based on the property of outdoor point clouds in autonomous driving scenarios, *i.e.*, the point clouds of distant objects are more sparse, we propose point density prediction to enable the 3D encoder to learn location information, which is essential for object detection. Experimental results show that BEV-MAE surpasses prior state-of-the-art self-supervised methods and achieves a favorably pre-training efficiency. Furthermore, based on TransFusion-L, BEV-MAE achieves new state-of-the-art LiDAR-based 3D object detection results, with 73.6 NDS and 69.6 mAP on the nuScenes benchmark. The source code will be released at https://github.com/VDIGPKU/BEV-MAE.

Introduction

3D object detection is one of the most basic tasks in autonomous driving. It aims to localize objects in 3D space and classify them simultaneously. In recent years, LiDAR-based 3D object detection methods have achieved significant success due to the increasing amount of labeled training data (Caesar et al. 2020). However, existing LiDAR-based 3D object detection methods for autonomous driving scenarios often adopt the paradigm of training from scratch, which brings two defects. First, the training-from-scratch paradigm largely relies on extensive labeled data. For 3D object detection, annotating precise bounding boxes and classification labels is costly and time-consuming, *e.g.*, it takes around $114s$ to annotate one object (Meng et al. 2020) on KITTI. Second, self-driving vehicles can generate massive unlabeled point cloud data daily in many practical scenarios, which cannot be used in the training-from-scratch paradigm.

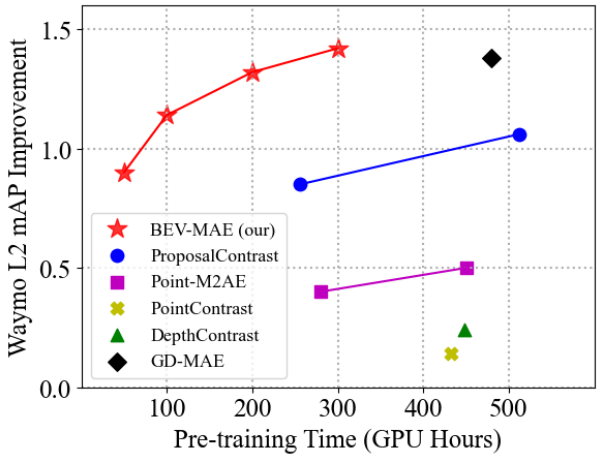

Figure 1: Performance improvement *vs.* Pre-training time trade-off. All entries are benchmarked by a P40 GPU. The 3D object detector is CenterPoint (Yin, Zhou, and Krahenbuhl 2021). All models are pre-trained on full Waymo and then fine-tuned with 20% training samples on Waymo.

A simple and desirable solution to the above two problems is self-supervised pre-training, widely used in fields such as computer vision (He et al. 2022) and natural language processing (Devlin et al. 2019). By solving pre-designed pretext tasks, self-supervised pre-training can learn a general and transferable feature representation on large-scale unlabeled data. One of the mainstream approaches in self-supervised learning is masked modeling, as illustrated in Fig. 2. Specifically, numerous approaches adopt masked image modeling (He et al. 2022; Xie et al. 2022) and masked language modeling (Devlin et al. 2019) to pre-train networks by reconstructing images, words, and sentences from masked inputs. Recently, several methods (Yu et al. 2022; Pang et al. 2022) use masked point modeling for dense point clouds, achieving promising performance on shape classification, shape segmentation, and indoor 3D object detection.

*Corresponding author.

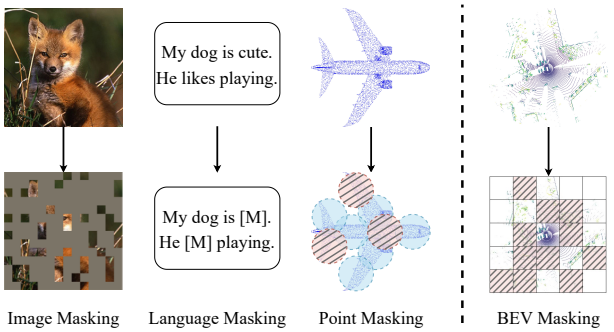

Figure 2: Illustration of several masking strategies in the masked modeling. MAE (He et al. 2022) masks non-overlapping image patches. BERT (Devlin et al. 2019) masks words or sentences. Point-MAE (Pang et al. 2022) uses furthest point sampling to create overlapping point patches. Our method (right) projects point clouds into a BEV plane, and masks points in non-overlapping BEV grids.

However, these schemes mainly focus on synthetic or indoor datasets, such as ShapeNet (Chang et al. 2015), ModelNet40 (Wu et al. 2015), and ScanNet (Dai et al. 2017). When applied to autonomous driving scenarios, where the range of scenes is more extensive and point cloud density is more sparse, their results are unsatisfactory (Liang et al. 2021). To address this issue, a few works (Hess et al. 2023; Tian et al. 2023) explore masked modeling for autonomous driving scenarios. However, these methods mainly adopt voxel-based masking strategies and recover masked points in voxels. There exist feature representation gaps between these *voxel-based* pre-training pipelines and the prevalent *BEV-based* 3D object detection methods.

In this work, we present bird's eye view masked autoencoders, dubbed BEV-MAE, specially for pre-training 3D object detectors on autonomous driving point clouds. Instead of randomly masking point clouds or voxels, we propose a BEV-guided masking strategy (right part of Fig. 2) for two benefits. First, we enforce the 3D encoder to learn feature representation in a BEV perspective by reconstructing masked information on the BEV plane. Therefore, during fine-tuning, the pre-trained 3D encoder can facilitate the training process of 3D detectors in the BEV perspective. Second, current 3D encoders of LiDAR-based 3D object detectors often downsample the resolution of points or voxels to save the computational overhead, *e.g.*, GPU memory, and training time. With the BEV-guided masking strategy, we do not need to design complicated decoders with upsampling operations (Shi et al. 2020) since the size of masked girds matches the resolution of BEV features. BEV-MAE can achieve promising results with a simple one-layer 3×3 convolution as the decoder. By designing the BEV-guided masking strategy, we obtain considerable pre-training efficiency improvement, as shown in the Fig. 1.

Since the commonly used sparse 3D convolutions only perform computation near the occupied areas, the receptive field size of the 3D encoder may become smaller with masked point inputs, resulting in low learning efficiency and poor transferability. To alleviate this issue, we replace the masked points with a shared learnable point token during pre-training to maintain a consistent receptive field size of the 3D encoder with fine-tuning. The shared learnable point token can help communication between unmasked areas without providing additional information to reduce the difficulty of the pre-training task. Moreover, we introduce point density prediction for masked areas in addition to using coordinates of masked point clouds as the reconstruction target (Hess et al. 2023; Yang et al. 2023). Since the point clouds become sparse when they are far from the LiDAR sensor in outdoor scenes, the density of point clouds can reflect the distance between points and the central LiDAR sensor. Naturally, density prediction can guide models to learn location information, which is critical for object detection.

The main contributions of this work are:

- We present a simple and efficient self-supervised pre-training method, BEV-MAE, tailored for LiDAR-based 3D object detectors in autonomous driving. With the proposed BEV-guided masking strategy, the 3D encoder of 3D object detectors can directly learn feature representation in a BEV perspective with a simple and lightweight decoder.
- We introduce a shared learnable point token to alleviate the inconsistency of the receptive field size for the 3D encoder during pre-training and fine-tuning, and propose density prediction to learn location information for the 3D encoder.
- BEV-MAE outperforms existing self-supervised methods in performance and pre-training efficiency. Moreover, BEV-MAE can further improve the performance of state-of-the-art 3D object detectors. Combined with TransFusion-L, BEV-MAE achieves new state-of-the-art results on nuScenes.

Related Work

3D Object Detection

The objective of 3D object detection is to localize objects of interest with 3D bounding boxes and classify the detected objects. Due to the large domain gap between indoor and outdoor datasets, the two cases' corresponding 3D object detection methods have been developing almost independently. Here we focus on the works about outdoor 3D object detection for autonomous driving. Recent outdoor 3D object detection approaches (Yan, Mao, and Li 2018; Yin, Zhou, and Krahenbuhl 2021; Lang et al. 2019) based on BEV representation attract much attention for their convenience in fusing different information, including multi-view (Huang et al. 2021), multi-modality (Liang et al. 2022), and temporal inputs (Huang and Huang 2022). To obtain BEV features, LiDAR-based methods first extract point features by a 3D encoder, such as PointPillar (Lang et al. 2019) and VoxelNet (Zhou and Tuzel 2018), and then project the features onto a BEV plane according to the point coordinates. For camera-based approaches, they extract the 2D features from multi-view images by a prevalent 2D backbone, including ResNet (He et al. 2016) and SwinTransformer (Liu et al. 2021), and utilize geometry-based view transforma-

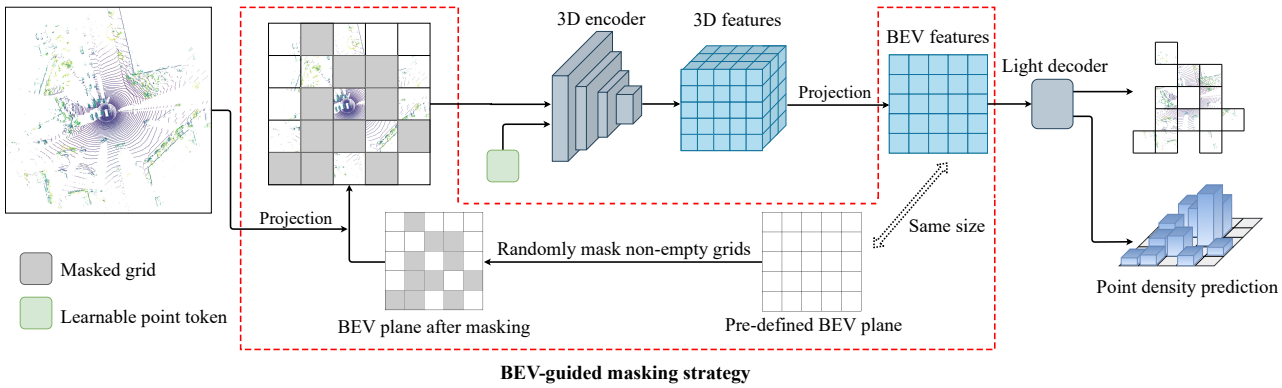

Figure 3: Overall pipeline of BEV-MAE. We first mask point clouds with the BEV-guided masking strategy. Then, the masked points are replaced with a shared learnable point token. After extracting BEV features by a 3D encoder from visible points, we send the features to a light decoder to reconstruct masked point clouds and predict the point density of masked grids.

tion (Philion and Fidler 2020) to construct BEV features from multi-view image features. Next, a 2D encoder and a detection head (Yan, Mao, and Li 2018; Yin, Zhou, and Krahenbuhl 2021) are applied for these methods to process the BEV features and predict the final detection results. In essence, this pipeline reduces the task of 3D object detection to 2D object detection. Hence, these methods can fully utilize the highly developed 2D object detection algorithms. Similarly, our work pre-trains the 3D encoder of 3D object detectors in the BEV space and can directly facilitate the 3D detection task in the BEV perspective.

Self-supervised Learning for Outdoor Point Clouds

The main idea of self-supervised learning is to train networks on unlabeled data with pretext tasks. Since no prior human knowledge is introduced, the pre-trained networks can learn a more general feature representation and have an excellent transfer ability. Recently, 3D representation learning of outdoor point clouds has attempted to adopt the core idea of self-supervised learning. GCC-3D (Liang et al. 2021) proposes geometry-aware contrast and harmonized cluster to learn geometry and semantic information from sparse point clouds. ProposalConstrast (Yin et al. 2022) samples region proposals for each point cloud with farthest point sampling (FPS) and jointly optimizes inter-proposal discrimination and inter-cluster separation. In addition to contrastive learning methods, masked modeling for outdoor point clouds has been studied recently for its simplicity and efficiency. Voxel-MAE (Hess et al. 2023) proposes a voxel-wised masking strategy for the transformer-based encoder. It reconstructs masked voxels and predicts the number of points and occupancy. GD-MAE (Yang et al. 2023) presents a multi-level transformer architecture and adopts a multi-scale masking strategy. It uses a generative decoder to recover masked patches with multi-scale features. GeoMAE (Tian et al. 2023) introduces pyramid centroid, occupancy, surface normal, and surface curvature of point clouds as prediction targets. MV-JAR (Xu et al. 2023) proposes a masked voxel Jigsaw and reconstruction method. MSP (Jiang et al. 2023) builds a masked shape prediction pipeline for 3D scene understanding. ALSO (Boulch et al. 2023) presents a query-based occupancy estimation to capture 3D semantic information. However, these works mainly adopt voxel-based masking strategies.

This paper proposes an efficient masked modeling framework, BEV-MAE, for pre-training point clouds in autonomous driving scenarios. Instead of masking voxels, BEV-MAE adopts a BEV-guided masking strategy to learn BEV feature representation and achieves new state-of-the-art self-supervised learning performance.

Method

The overall pipeline of BEV-MAE is shown in Fig. 3. BEV-MAE first uses a BEV-guided masking strategy to mask point clouds. These masked points are then replaced with a shared learnable point token. We send the processed points into a 3D encoder and a light decoder sequentially. Finally, the light decoder will reconstruct the masked points and predict the point density of the masked area.

BEV-guided Masking Strategy

In LiDAR-based 3D object detection, the point clouds are often divided into regular voxels. A straightforward masking strategy is to mask the voxels like masking patches in vision (He et al. 2022; Xie et al. 2022). However, this simple voxel masking strategy makes subsequent decoder design difficult. In addition, it takes little consideration of the type of feature representation in the mainstream 3D object detection methods for autonomous driving, *i.e.*, BEV feature representation. To this end, we propose a BEV-guided masking strategy to mask points in the BEV plane.

Specifically, assuming the resolution of the features in the BEV perspective after encoding and transformation is $X \times Y \times C$, we first pre-define a grid-shaped BEV plane with the size of $X \times Y$. We then project each LiDAR point p_k into a corresponding BEV grid $g_{i,j}$ of the pre-defined BEV plane according to its point coordinates (x_{p_k}, y_{p_k}). Each BEV grid will contain a various number of points:

$$g_{i,j} = \{p_k \mid \lfloor x_{p_k}/d \rfloor = i,\ \lfloor y_{p_k}/d \rfloor = j\}, \qquad (1)$$

where d is the downsample ratio of the 3D encoder and $\lfloor x \rfloor$ denotes rounding down of x. We randomly select a significant fraction of non-empty BEV grids, *i.e.*, $g_{i,j} \neq \varnothing$, as the masked grids $\{g_i^m\}$ and denote the remaining BEV grids as visible grids $\{g_i^v\}$. Finally, we obtain the visible and masked point clouds by merging the points in $\{g_i^m\}$ and $\{g_i^v\}$, formulated as $\{p_k^v\} = \bigcup_i g_i^v$ and $\{p_k^m\} = \bigcup_i g_i^m$ respectively.

Learnable Point Token

The 3D encoder of recent voxel-based 3D object detectors typically consists of several sparse convolution operations, which only process the features near the non-empty voxels. When only taking visible point clouds $\{p_k^v\}$ as input, the size of the receptive field of the 3D encoder becomes smaller. To address this issue, we replace the masked point clouds $\{p_k^m\}$ with a shared learnable point token. Specifically, we use coordinates of full point clouds as the input indexes (Yan, Mao, and Li 2018) of sparse convolution and replace the feature of masked point clouds with the shared learnable point token in the first sparse convolution layer. We keep the other sparse convolutional layers unchanged. The goal of the proposed shared learnable point token is to pass the information from one point or voxel to another to maintain the size of the receptive field. It does not introduce any additional information, including the coordinates of masked points.

Decoder Design

In the BEV-guided masking strategy, the size of masked areas is aligned with the resolution of the BEV features. Thus, we can directly predict the reconstruction results of one masked grid from the corresponding BEV features without upsampling operations.

The decoder of BEV-MAE is only used during pre-training to solve the masking task. Naturally, the design of the decoder is flexible and independent of the 3D encoder architecture. We evaluate three types of decoders, *i.e.*, one-layer 3×3 convolution, transformer block (He et al. 2022), and residual convolution block (He et al. 2016), and find the highly lightweight decoder, a one-layer 3×3 convolution, can achieve impressive performance while reducing pre-training time and GPU memory cost.

Reconstruction Target

The proposed BEV-MAE is supervised by two tasks, *i.e.*, point cloud reconstruction, and density prediction. A separate linear layer is applied as the prediction head for each task to predict results. We describe each task below.

Point Cloud Reconstruction. BEV-MAE reconstructs the masked input by predicting the coordinates of masked points. However, each masked grid in the pre-defined BEV plane contains a different number of points, leading to the challenges of designing the prediction head.

To address this issue, we propose to reconstruct the local structure of the masked point clouds with a set-to-set prediction. Specifically, we apply a linear layer to predict a set of 3D points with a fixed number, denoted as $P_i = \{p_l \mid l = 1, 2, ..., L\}$, for each masked grid g_i^m. Given the original points $\hat{P}_i = \{\hat{p_k} \mid k = 1, 2, ..., N\}$ in g_i^m, where N varies with the grid, we utilize Chamfer Distance (Fan, Su, and Guibas 2017) between predictions P_i and the ground-truth $\hat{P}_i$ as the reconstruction loss:

$$\mathcal{L}_c^i = \frac{1}{L} \sum_{p_l \in P_i} \min_{\hat{p_k} \in \hat{P}_i} ||p_l - \hat{p_k}||_2^2 + \frac{1}{N} \sum_{\hat{p_k} \in \hat{P}_i} \min_{p_l \in P_i} ||\hat{p_k} - p_l||_2^2. \tag{2}$$

We then average the loss over all the masked grids as the final reconstruction loss:

$$\mathcal{L}_c = \frac{1}{n_m} \sum_{i=1}^{n_m} \mathcal{L}_c^i, \tag{3}$$

where n_m is the number of masked grids. The Chamfer distance measures the distance between two sets with different cardinalities. Thus, it enforces the shape of predicted point clouds to mimic the local structure of masked inputs (Fan, Su, and Guibas 2017).

Since the coordinate value of each ground-truth point varies largely in the different grids, directly predicting the absolute coordinates of each point may cause instability during pre-training. To alleviate this issue, we predict the normalized coordinate in point cloud reconstruction. Specifically, we first calculate the coordinate offset of each ground-truth point to the center of its corresponding BEV grid and then normalize the offset value by the size of the BEV grid.

Point Density Prediction. Unlike image, language, and indoor point clouds, outdoor point clouds in autonomous driving scenarios have the property that the density of the point clouds becomes small when they are far from the LiDAR sensor. Consequently, the density can reflect the location of each point or object. Furthermore, for object detection, the localization ability of detectors is essential. Based on the above analysis, we propose another task for BEV-MAE, *i.e.*, point density prediction, to guide the 3D encoder to achieve a better localization ability. Compared with predicting the number of points in Voxel-MAE (Hess et al. 2023), our proposed density prediction is more stable and effective during training.

For each masked grid g_i^m, we count the number of points in this grid and calculate the density $\hat{\rho}_i$ by dividing the number of points with the occupied volume in 3D space as the ground truth for density prediction. Then, we use a linear layer as the prediction head to obtain the density prediction ρ_i. We supervise this task with the *Smooth*-ℓ_1 loss:

$$\mathcal{L}_d^i = Smooth\text{-}\ell_1(\rho_i - \hat{\rho}_i). \tag{4}$$

Similarly, we average the loss over all the masked grids as the final density prediction loss:

$$\mathcal{L}_d = \frac{1}{n_m} \sum_{i=1}^{n_m} \mathcal{L}_d^i. \tag{5}$$

Experimental Results

Implementation Details

We evaluate the proposed BEV-MAE on two popular large-scale autonomous driving datasets, *i.e.*, nuScenes and

Pre-traing Method	Epochs	Time	Dataset fraction	L2 (mAP/APH)			
				Overall	Vehicle	Pedestrian	Cyclist
From-scratch	-	-	-	65.60 / 63.21	64.18 / 63.69	65.22 / 59.68	67.41 / 66.25
GCC-3D (Liang et al. 2021)*	40	-	100%	65.29 / 62.79	63.97 / 63.47	64.23 / 58.47	67.68 / 66.44
PointContrast (Xie et al. 2020)	50	54h	100%	$65.88^{+0.28}$ / $63.28^{+0.07}$	63.81 / 63.33	66.67 / 60.51	67.17 / 66.00
DepthContrast (Zhang et al. 2021)	50	56h	100%	$65.84^{+0.24}$ / $63.33^{+0.12}$	64.45 / 63.95	65.61 / 59.86	67.43 / 66.22
Point-M2AE (Zhang et al. 2022)	30	56h	100%	$66.10^{+0.50}$ / $63.59^{+0.38}$	64.26 / 63.77	65.64 / 60.00	68.20 / 67.01
ProposalContrast (Yin et al. 2022)	50	64h	100%	$66.42^{+0.82}$ / $63.85^{+0.64}$	65.03 / 64.53	65.93 / 59.95	68.26 / 67.04
MSP (Jiang et al. 2023)	30	-	100%	- / $64.26^{+1.05}$	- / -	- / -	- / -
GD-MAE† (Yang et al. 2023)	30	60h	100%	$66.98^{+1.38}$ / $64.53^{+1.32}$	65.64 / 64.95	66.39 / 61.12	68.92 / 67.52
BEV-MAE (Ours)	20	5h	20%	$66.70^{+1.10}$ / $64.25^{+1.04}$	64.71 / 64.22	66.21 / 60.59	69.11 / 67.93
BEV-MAE (Ours)	30	38h	100%	$\mathbf{67.02^{+1.42}}$ / $\mathbf{64.55^{+1.34}}$	65.01 / 64.53	66.58 / 60.87	69.46 / 68.25

Table 1: Comparisons between BEV-MAE and state-of-the-art self-supervised learning methods on Waymo validation set. All detectors are fine-tuning with 20% training samples on Waymo following the OpenPCDet configuration. Here, the entry with * denotes the results are from the paper (Liang et al. 2021); the entry with † indicates the results are implemented by the released official code[1]. 'Epochs' indicates the pre-training epochs; 'Dataset fraction' means the data fraction of the Waymo training set used for pre-training; and 'Time' refers to the pre-training time estimated by 8 P40 GPU.

Method	NDS	mAP
CenterPoint (Yin, Zhou, and Krahenbuhl 2021)	65.5	58.0
VISTA (Deng et al. 2022)	69.8	63.0
FocalsConv (Chen et al. 2022)	70.0	63.8
VoxelNeXt (Chen et al. 2023b)	70.0	64.5
TransFusion-L (Bai et al. 2022)	70.2	65.5
LargeKernel3D (Chen et al. 2023a)	70.6	65.4
Link (Lu et al. 2023)	71.0	66.9
GeoMAE†(Tian et al. 2023)	72.5	67.8
BEV-MAE (Ours)	71.7	67.0
BEV-MAE† (Ours)	**73.6**	**69.6**

Table 2: Performances of 3D object detection on the nuScenes *test* split. † results are obtained using a modified model structure.

Waymo Open Dataset. We mainly focus on the evaluation metrics of mAP and NDS for nuScenes and the more difficult LEVEL_2 metric (L2 mAP and L2 APH) for Waymo.

During pre-training, We train BEV-MAE with the Adam optimizer under the one-cycle schedule. The maximum learning rate is 0.0003 with a batch size of 4. The masking ratio is 0.7. The number of predicted points L in point cloud reconstruction is 20 for each masked grid. For nuScenes, we use a common strategy (Yin, Zhou, and Krahenbuhl 2021) that concatenates one labeled point cloud frame and nine consecutive unlabeled point cloud sweeps to form a denser point cloud input.

Waymo Results

On the Waymo dataset, we adopt VoxelNet as the 3D encoder and use CenterPoint as the LiDAR-based object detector following ProposalContrast. We replace the multi-scale encoder of GD-MAE with VoxelNet following MV-JAR for fair comparisons. Table 1 shows that BEV-MAE substantially outperforms the training-from-scratch baselines and state-of-the-art self-supervised learning methods. Specifically, BEV-MAE outperforms the training-from-scratch baseline by 1.42 mAP and 1.34 APH, outperforming ProposalContrast by 0.60 mAP and 0.70 APH. BEV-MAE outperforms GA-MAE with 63% pre-training cost. In addition, we observe that, with only 20% training data and 7% computation cost, BEV-MAE achieves better results compared to ProposalContrast with 100% data for pre-training. Meanwhile, BEV-MAE achieves comparable results to MSP with 20% pre-training data. The results demonstrate the efficiency of BEV-MAE pre-training.

NuScenes Results

For the nuScenes dataset, we apply BEV-MAE to a strong 3D object detector, *i.e.,* TransFusion-L. Table 2 compares BEV-MAE with several LiDAR-based 3D object detectors on the nuScenes *test* split. BEV-MAE outperforms TransFusion-L baseline by 1.5 NDS and 1.5 mAP, surpassing Link by 0.7 NDS. Notably, GeoMAE uses a multi-stride structure to improve detection performance further. To compare with GeoMAE, we increase the number of channels and layers for the convolution block in the 3D encoder. BEV-MAE outperforms GeoMAE and achieves new state-of-the-art results with 73.6 NDS and 69.6 mAP.

Data Efficiency

To assess the data efficiency of BEV-MAE, we train the 3D detectors with different amounts of labeled data. We follow the settings of the data-efficient benchmark proposed by MV-JAR (Xu et al. 2023) to split the fine-tuning dataset and train the detectors. As shown in Table 3, pre-training with BEV-MAE can consistently improve the detection results on different fractions of the training data. Especially,

[1]https://github.com/Nightmare-n/GD-MAE

Data amount	Initialization	Overall		Car		Pedestrian		Cyclist	
		L2 mAP	L2 mAPH	L2 mAP	L2 mAPH	L2 mAP	L2 mAPH	L2 mAP	L2 mAPH
5%	Random	44.41	40.34	51.01	50.49	52.74	42.26	29.49	28.27
	PointContrast(Xie et al. 2020)	45.32	41.30	52.12	51.61	53.68	43.22	30.16	29.09
	ProposalContrast(Yin et al. 2022)	46.62	42.58	52.67	52.19	54.31	43.82	32.87	31.72
	MV-JAR (Xu et al. 2023)	50.52	46.68	56.47	56.01	57.65	47.69	37.44	36.33
	BEV-MAE (Ours)	**51.63**	**47.77**	56.35	55.81	58.11	48.37	40.44	39.13
10%	Random	54.31	50.46	54.84	54.37	60.55	50.71	47.55	46.29
	PointContrast(Xie et al. 2020)	53.69	49.94	54.76	54.30	59.75	50.12	46.57	45.39
	ProposalContrast(Yin et al. 2022)	53.89	50.13	55.18	54.71	60.01	50.39	46.48	45.28
	MV-JAR (Xu et al. 2023)	57.44	54.06	58.43	58.00	63.28	54.66	50.63	49.52
	BEV-MAE (Ours)	**58.16**	**54.75**	58.51	57.94	63.83	55.23	52.13	51.07
20%	Random	60.16	56.78	58.79	58.35	65.63	57.04	56.07	54.94
	PointContrast(Xie et al. 2020)	59.35	55.78	58.64	58.18	64.39	55.43	55.02	53.73
	ProposalContrast(Yin et al. 2022)	59.52	55.91	58.69	58.22	64.53	55.45	55.36	54.07
	MV-JAR (Xu et al. 2023)	62.28	59.15	61.88	61.45	66.98	59.02	57.98	57.00
	BEV-MAE (Ours)	**62.88**	**59.97**	61.79	61.37	67.35	59.39	59.51	59.14
50%	Random	66.43	63.36	63.81	63.38	70.78	63.05	64.71	63.66
	PointContrast(Xie et al. 2020)	65.51	62.21	62.66	62.23	69.82	61.53	64.04	62.86
	ProposalContrast(Yin et al. 2022)	65.76	62.49	62.93	62.50	70.09	61.86	64.26	63.11
	MV-JAR (Xu et al. 2023)	66.70	63.69	64.30	63.89	71.14	63.57	64.65	63.63
	BEV-MAE (Ours)	**67.16**	**64.07**	64.33	63.84	71.38	63.61	65.76	64.77
100%	Random	68.50	65.54	64.96	64.56	72.38	64.89	68.17	67.17
	PointContrast(Xie et al. 2020)	68.06	64.84	64.24	63.82	71.92	63.81	68.03	66.89
	ProposalContrast(Yin et al. 2022)	68.17	65.01	64.42	64.00	71.94	63.94	68.16	67.10
	MV-JAR (Xu et al. 2023)	69.16	66.20	65.52	65.12	72.77	65.28	69.19	68.20
	BEV-MAE (Ours)	**69.35**	**66.46**	65.54	65.02	72.84	65.31	69.67	69.05

Table 3: Results about data efficiency on Waymo. The detectors are fine-tuned on various fractions of Waymo training split following MV-JAR (Xu et al. 2023). 'Random' denotes the training-from-scratch baseline.

our method can obtain more significant gains when the labeled data is less. For example, BEV-MAE surpasses the training-from-scratch baseline by 7.22 mAP and 7.43 APH on L2 when using 5% labeled data. These results suggest the potential of BEV-MAE in using large amounts of unlabeled data. Furthermore, we observe that BEV-MAE brings marginal improvements when fine-tuned on the full Waymo dataset. The reason is that the detector's capacity becomes the bottleneck for detection performance (Xu et al. 2023).

Transfer Learning

We evaluate the cross-dataset transferability of the pre-trained encoder by fine-tuning it on a different dataset. Since the collection of different datasets uses various types of sensors, we only use the coordinates as the input feature of point clouds for compatibility across datasets. Table 4 shows that BEV-MAE consistently improves the detection performance across different datasets. In addition, we find that CenterPoint achieves better results when pre-training and fine-tuning on the same dataset. The domain gaps between two datasets likely affect the transfer performance negatively, *e.g.*, the density of point clouds on Waymo is five times higher than it of nuScenes.

Moreover, we conduct a more realistic pre-training setting, *i.e.*, pre-trained on a combined dataset and fine-tuned on the target dataset. The performance of the 3D object detector on Waymo and nuScenes can be further improved under this setting. These results indicate that BEV-MAE can exploit existing various datasets for better pre-training.

Pre-train \ Fine-tune	nuScenes		Waymo	
	mAP	NDS	L2 mAP	L2 APH
Random init.	48.6	58.4	63.97	61.53
nuScenes	$49.7^{+1.1}$	$58.9^{+0.5}$	$64.79^{+0.82}$	$62.28^{+0.75}$
Waymo	$49.4^{+0.8}$	$58.8^{+0.4}$	$65.13^{+1.16}$	$62.63^{+1.10}$
nuScenes + Waymo	$\mathbf{50.1^{+1.5}}$	$\mathbf{59.1^{+0.7}}$	$\mathbf{65.36^{+1.39}}$	$\mathbf{62.89^{+1.36}}$

Table 4: Results of transfer learning. The contents in the column and row show the datasets for pre-training and fine-tuning, respectively.

Ablation Study

We conduct ablation experiments to analyze the effectiveness of each setting of BEV-MAE, including the main components and hyper-parameters. In the following experiments, we use CenterPoint as the 3D detector. We first pre-

Pre-train	Reconstruction target	LT	L2 mAP	L2 APH
None	-	-	65.60	63.21
BEV-MAE	Coord. (w/o norm)	✓	65.66	63.09
	Coord. (w norm)	✓	66.20	63.71
	Density	✓	65.80	63.27
	Number of points	✓	65.32	62.88
	Coord. (w norm) + Density	×	66.49	63.99
	Coord. (w norm) + Density	✓	**66.70**	**64.25**

Table 5: Ablation on main components. 'LT' denotes the shared learnable point token. Each component brings performance improvement for BEV-MAE.

train the 3D encoder on 20% training data of Waymo with BEV-MAE and then evaluate its performance by fine-tuning with CenterPoint on the same 20% training data.

Main Component. We evaluate the effectiveness of each component of the proposed BEV-MAE, as shown in Table 5. Pre-training with normalized coordinates or density prediction can improve 3D detection performance. The detection results can be further improved by taking both coordinate and density prediction as the pre-training task, achieving 66.49 mAP and 63.99 APH. For coordinate prediction, the normalization operation to process the coordinates of points is essential.

Replacing density prediction with the number of points prediction in Voxel-MAE (Hess et al. 2023) decreases the detection performance. The reason is that since the number of points in a BEV grid varies from one to hundreds, the pre-training process is unstable when using the number of points prediction as the target. Moreover, pre-training with the shared learnable token brings additional performance gain of 0.21 mAP and 0.26 APH.

Decoder Design. In addition to one-layer 3×3 convolution, we also test two decoders with more complex design, *i.e.*, Transformer block (He et al. 2022) and Residual Conv block (He et al. 2016), as shown in Table 6. The results show that the detection performance drops with more complex decoders. The performance decreases more when using the Transformer block as the decoder, which we assign to the different architecture between the encoder and decoder (Sparse Convolution *vs.* Transformer). In addition, we observe that the complex decoders bring additional training cost, *e.g.*, Residual Conv block brings additional 0.2× training cost compared with one-layer 3×3 Conv. These results show that exploring decoder design may not be essential for BEV-MAE. A simple one-layer 3×3 convolution is adequate for practice.

Masking Strategy. We study how different masking strategies, including a simple voxel masking strategy and the proposed BEV-guided masking strategy, affect the effectiveness of representation learning. For the voxel masking strategy, we randomly mask non-empty voxels and use sparse deconvolution layers (Shi et al. 2020) as the decoder to recover the points in masked voxels. In Table 7, we observe

Decoder	L2 mAP	L2 APH	Training cost
One-layer 3×3 Conv	**66.70**	**64.25**	**1**×
Residual Conv block	66.61	64.09	1.2×
Transformer block	65.80	63.26	1.4×

Table 6: Ablation on different deocders. One-layer 3×3 Conv achieves the best results with the least training cost.

Masking strategy	L2 mAP	L2 APH	Memory	Training cost
BEV-guided masking	**66.70**	**64.25**	**4.1G**	**1**×
Voxel masking	66.63	64.16	12.6G	1.4×

Table 7: Ablation on the masking strategy. Pre-training with the BEV-guided masking strategy performs better with less GPU memory consumption and pre-training cost.

Masking ratio	L2 mAP	L2 APH
50%	66.45	64.00
60%	66.62	64.13
70%	**66.70**	**64.25**
80%	66.52	64.06

Table 8: Ablation on masking ratio. The fine-tuning results are less sensitive to the masking ratio.

that the proposed BEV-guided masking strategy achieves better transfer performance on downstream 3D object detection tasks. In addition, compared with the voxel masking strategy, the BEV-guided masking strategy significantly reduces the GPU memory consumption (4.1G *vs.* 12.6G) and training costs (1× *vs.* 1.4×) during pre-training.

Masking Ratio. Table 8 ablates the influence of the masking ratio. The proposed BEV-MAE works well within a wide range of masking ratios (50%-80%). The best results are achieved when the masking ratio is 70%.

Conclusions

In this work, we address the problem of self-supervised pre-training on point clouds in autonomous driving scenarios. We present BEV-MAE to pre-train the 3D encoder of LiDAR-based 3D object detectors. Instead of simply masking points or voxels, we propose a BEV-guided masking strategy for better BEV representation learning and to avoid complex decoder design. Furthermore, we introduce a shared learnable point token to maintain the receptive field size of the encoder during pre-training and fine-tuning. By leveraging the properties of outdoor point clouds in autonomous driving scenarios, we propose point density prediction to guide the encoder to learn location information. Experimental results show that BEV-MAE surpasses previous self-supervised learning methods in performance and pre-training efficiency. Moreover, BEV-MAE can further boost the performance of the state-of-the-art 3D object detectors and achieve new state-of-the-art results on nuScenes.

Acknowledgements

This work was supported by National Key R&D Program of China (Grant No. 2022ZD0160305).

References

Bai, X.; Hu, Z.; Zhu, X.; Huang, Q.; Chen, Y.; Fu, H.; and Tai, C.-L. 2022. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *CVPR*.

Boulch, A.; Sautier, C.; Michele, B.; Puy, G.; and Marlet, R. 2023. ALSO: Automotive Lidar Self-Supervision by Occupancy Estimation. In *CVPR*.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuScenes: A Multimodal Dataset for Autonomous Driving. In *CVPR*.

Chang, A. X.; Funkhouser, T. A.; Guibas, L. J.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; Xiao, J.; Yi, L.; and Yu, F. 2015. ShapeNet: An Information-Rich 3D Model Repository. arXiv:1512.03012.

Chen, Y.; Li, Y.; Zhang, X.; Sun, J.; and Jia, J. 2022. Focal Sparse Convolutional Networks for 3D Object Detection. In *CVPR*.

Chen, Y.; Liu, J.; Zhang, X.; Qi, X.; and Jia, J. 2023a. LargeKernel3D: Scaling up Kernels in 3D Sparse CNNs. In *CVPR*.

Chen, Y.; Liu, J.; Zhang, X.; Qi, X.; and Jia, J. 2023b. VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking. In *CVPR*.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*.

Deng, S.; Liang, Z.; Sun, L.; and Jia, K. 2022. Vista: Boosting 3d object detection via dual cross-view spatial attention. In *CVPR*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Fan, H.; Su, H.; and Guibas, L. J. 2017. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. In *CVPR*.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. B. 2022. Masked Autoencoders Are Scalable Vision Learners. In *CVPR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Hess, G.; Jaxing, J.; Svensson, E.; Hagerman, D.; Petersson, C.; and Svensson, L. 2023. Masked autoencoder for self-supervised pre-training on lidar point clouds. In *WACV*.

Huang, J.; and Huang, G. 2022. BEVDet4D: Exploit Temporal Cues in Multi-camera 3D Object Detection. arXiv:2203.17054.

Huang, J.; Huang, G.; Zhu, Z.; and Du, D. 2021. BEVDet: High-performance Multi-camera 3D Object Detection in Bird-Eye-View. arXiv:2112.11790.

Jiang, L.; Yang, Z.; Shi, S.; Golyanik, V.; Dai, D.; and Schiele, B. 2023. Self-supervised Pre-training with Masked Shape Prediction for 3D Scene Understanding. In *CVPR*.

Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*.

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 *ICCV*.

Liang, T.; Xie, H.; Yu, K.; Xia, Z.; Lin, Z.; Wang, Y.; Tang, T.; Wang, B.; and Tang, Z. 2022. BEVFusion: A Simple and Robust LiDAR-Camera Fusion Framework. In *NeurIPS*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*.

Lu, T.; Ding, X.; Liu, H.; Wu, G.; and Wang, L. 2023. LinK: Linear Kernel for LiDAR-Based 3D Perception. In *CVPR*.

Meng, Q.; Wang, W.; Zhou, T.; Shen, J.; Van Gool, L.; and Dai, D. 2020. Weakly supervised 3d object detection from lidar point cloud. In *ECCV*.

Pang, Y.; Wang, W.; Tay, F. E. H.; Liu, W.; Tian, Y.; and Yuan, L. 2022. Masked Autoencoders for Point Cloud Self-supervised Learning. In *ECCV*.

Philion, J.; and Fidler, S. 2020. Lift, Splat, Shoot: Encoding Images from Arbitrary Camera Rigs by Implicitly Unprojecting to 3D. In *ECCV*.

Shi, S.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE TPAMI*.

Tian, X.; Ran, H.; Wang, Y.; and Zhao, H. 2023. GeoMAE: Masked Geometric Target Prediction for Self-Supervised Point Cloud Pre-Training. In *CVPR*.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3D ShapeNets: A deep representation for volumetric shapes. In *CVPR*.

Xie, S.; Gu, J.; Guo, D.; Qi, C. R.; Guibas, L. J.; and Litany, O. 2020. PointContrast: Unsupervised Pre-training for 3D Point Cloud Understanding. In *ECCV*.

Xie, Z.; Zhang, Z.; Cao, Y.; Lin, Y.; Bao, J.; Yao, Z.; Dai, Q.; and Hu, H. 2022. SimMIM: a Simple Framework for Masked Image Modeling. In *CVPR*.

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 *CVPR*.

Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. *Sensors*.

Yang, H.; He, T.; Liu, J.; Chen, H.; Wu, B.; Lin, B.; He, X.; and Ouyang, W. 2023. GD-MAE: Generative Decoder for MAE Pre-Training on LiDAR Point Clouds. In *CVPR*.

Yin, J.; Zhou, D.; Zhang, L.; Fang, J.; Xu, C.; Shen, J.; and Wang, W. 2022. ProposalContrast: Unsupervised Pre-training for LiDAR-Based 3D Object Detection. In *ECCV*.

Yin, T.; Zhou, X.; and Krahenbuhl, P. 2021. Center-based 3d object detection and tracking. In *CVPR*.

Yu, X.; Tang, L.; Rao, Y.; Huang, T.; Zhou, J.; and Lu, J. 2022. Point-BERT: Pre-training 3D Point Cloud Transformers with Masked Point Modeling. In *CVPR*.

Zhang, R.; Guo, Z.; Gao, P.; Fang, R.; Zhao, B.; Wang, D.; Qiao, Y.; and Li, H. 2022. Point-M2AE: Multi-scale Masked Autoencoders for Hierarchical Point Cloud Pretraining. arXiv:2205.14401.

Zhang, Z.; Girdhar, R.; Joulin, A.; and Misra, I. 2021. Self-Supervised Pretraining of 3D Features on any Point-Cloud. In *ICCV*.

Zhou, Y.; and Tuzel, O. 2018. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *CVPR*.