HAL open science

OccFeat: Self-supervised Occupancy Feature Prediction for Pretraining BEV Segmentation Networks

Sophia Sirko-Galouchenko, Alexandre Boulch, Spyros Gidaris, Andrei Bursuc, Antonin Vobecky, Patrick Pérez, Renaud Marlet

To cite this version:

Sophia Sirko-Galouchenko, Alexandre Boulch, Spyros Gidaris, Andrei Bursuc, Antonin Vobecky, et al.. OccFeat: Self-supervised Occupancy Feature Prediction for Pretraining BEV Segmentation Networks. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Jun 2024, Vancouver, Canada. pp.4493-4503, ⟨10.1109/CVPRW63382.2024.00452⟩. ⟨hal-05078857⟩

HAL Id: hal-05078857

https://hal.science/hal-05078857v1

Submitted on 23 May 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

arXiv:2404.14027v3 [cs.CV] 12 Jun 2024

OccFeat: Self-supervised Occupancy Feature Prediction for Pretraining BEV Segmentation Networks

Sophia Sirko-Galouchenko[1,2] Alexandre Boulch[1] Spyros Gidaris[1] Andrei Bursuc[1]
Antonin Vobecky[1,3,4] Patrick Pérez[5*] Renaud Marlet[1,6]

[1]Valeo.ai, Paris, France [2]ISIR, Sorbonne Université, Paris, France [3]FEE CTU[†], Prague, Czechia
[4]CIIRC CTU, Prague, Czechia [5]Kyutai, Paris, France
[6]LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-Vallée, France

Abstract

We introduce a self-supervised pretraining method, called OccFeat, for camera-only Bird's-Eye-View (BEV) segmentation networks. With OccFeat, we pretrain a BEV network via occupancy prediction and feature distillation tasks. Occupancy prediction provides a 3D geometric understanding of the scene to the model. However, the geometry learned is class-agnostic. Hence, we add semantic information to the model in the 3D space through distillation from a self-supervised pretrained image foundation model. Models pretrained with our method exhibit improved BEV semantic segmentation performance, particularly in low-data scenarios. Moreover, empirical results affirm the efficacy of integrating feature distillation with 3D occupancy prediction in our pretraining approach.

1. Introduction

Camera-only bird's-eye-view (BEV) networks have gained significant interest in recent years within the field of autonomous driving perception [6, 24, 31, 34, 45, 58, 83]. The appeal of the BEV, or else top-view, is that it offers a unified space for various sensors, including surround-view cameras, Lidar, and radar [29, 51, 53], for both annotation and runtime perception purposes, and can serve as input for subsequent tasks in the driving pipeline, such as forecasting and planning [1, 8, 29, 31, 32]. Common tasks in the BEV space are semantic segmentation of objects [6, 24, 58, 83] and layouts [39, 40], as well as object detection [34, 45]. Our work specifically targets the camera-only BEV semantic segmentation task.

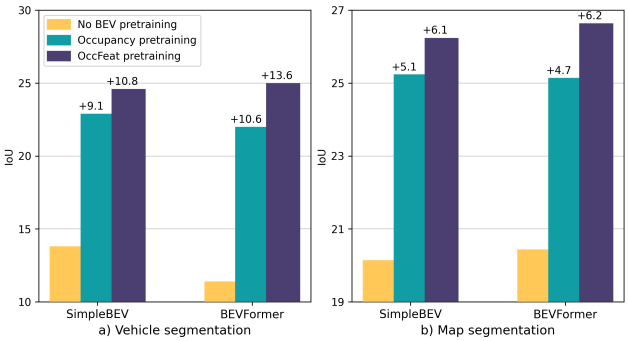


Figure 1. Performance comparison in low data regime (1% annotated data of nuScenes)

Until now, training networks for camera-only semantic segmentation in BEV space has relied on full supervision, necessitating annotations for each scene. This process is time-consuming due to the transition from the input image to the "synthetic" BEV space. For instance, annotations are typically generated for Lidar data, checked for visibility and classes in images, and then projected onto BEV segmentation images. To reduce annotation costs, we explore the potential of self-supervised pretraining camera-only BEV segmentation networks.

In self-supervised pretraining, networks are typically trained on annotation-free pretext tasks before the primary (downstream) task, like semantic segmentation. The aim is to guide the network in learning useful data representations during this pretraining phase. This process is intended to enhance the network's performance on the downstream task, enabling it to achieve higher accuracy while utilizing a reduced amount of annotated data. Pretraining has been proven efficient for several modalities from images [15, 27] to Lidar [7] and with different strategies, such as contrastive learning [15, 26], teacher-student architectures [12, 21–23] or reconstruction tasks [4, 27].

*Work done at valeo.ai.
[†]Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague.

In the field of autonomous driving, self-supervised pretraining for camera-only BEV networks has received limited attention despite its crucial role. Recently, a few methods have emerged that delve into this subject [54, 79, 81], but they predominantly focus on pretraining with 3D geometry prediction tasks. For instance, ViDAR [81] employs Lidar point cloud forecasting for pre-training, UniPAD [79] 3D surface and RGB pixels reconstruction, and UniScene [54] 3D occupancy prediction. While these methods equip the BEV networks with 3D geometry understanding, they often fall short in making the network capture semantic-aware information of the 3D scene, essential for tasks like BEV-based semantic segmentation.

Our approach, called Occupancy Feature Prediction (OccFeat), addresses this gap by presenting a pretraining objective that promotes a more comprehensive understanding of the 3D scene, encompassing both geometric and semantic aspects. In our approach, the camera-only BEV network is tasked to predict a 3D voxel-grid representation that includes (a) features indicating voxel occupancy and (b) high-level self-supervised image features characterizing occupied voxels.

To create this target voxel grid representation, we leverage aligned Lidar and image data in autonomous driving setups, along with a self-supervised image foundation model like DINOv2 [56], which has been pretrained to extract high-level 2D image features. Specifically, the occupancy of each voxel is determined using Lidar data, considering a voxel occupied if it contains at least one Lidar point. Simultaneously, the self-supervised image foundation model fills the occupied voxels with high-level image features. This process involves projecting the center coordinates of each occupied voxel cell into the 2D space of the image features extracted from the foundation model.

Unlike approaches solely focused on 3D geometry prediction, our method goes beyond by training the BEV network to predict a richer, more semantic representation of the 3D scene, all without requiring manual annotation, leveraging the pre-trained image foundation model. We empirically demonstrate that this enhancement leads to significantly better downstream BEV semantic segmentation results, especially in low-data regimes (e.g., see Fig. 1).

Our contributions are the following:

1. We present OccFeat, a self-supervised pretraining approach for camera-only BEV segmentation networks that enforces both geometric and semantic understanding of 3D scenes.
2. OccFeat exploits three modalities for pretraining: image, Lidar, and DINOv2 features. To the best of our knowledge, we are the first to leverage foundation image models (DINOv2) for pretraining camera-only BEV networks. We note that after pretraining, the Lidar and DINOv2 data are not used anymore.
3. We evaluate OccFeat on nuScenes [9] for BEV semantic segmentation of both vehicles and map layout. The results show the benefit of our pretraining method, especially in low-shot regimes, e.g., when using annotations only for 1% or 10% of nuScene's training data. Additionally, our OccFeat pretraining improves the robustness, as evaluated on the nuScenes-C benchmark [74].

2. Related work

Camera-only BEV perception. BEV perception aims for a unified representation of the surrounding environment of a vehicle. BEV has recently arised as a prevailing paradigm for multi-camera perception systems for autonomous driving. Camera-based BEV models are typically composed of three parts: (i) an image encoder shared across cameras for extracting 2D features, (ii) a view-projection module for "lifting" features in the 3D space to produce BEV features, and (iii) one or more task decoder modules that process BEV features towards addressing a task of interest, e.g., semantic segmentation, map prediction, 3D detection, etc. Among them, the view-projection has been in the spotlight of numerous works in this area in the past few years. The diversity of the approaches for view projection is impressive, ranging from purely geometric ones, e.g., using inverse perspective mapping with strong assumptions about the world [63] to projections completely learned from data, typically leveraging a cross-attention mechanism between image features (often imbued with 3D knowledge or priors) and learnable-queries from the projected space [6, 48, 49, 75, 83]. Currently, the most commonly used ones belong to one of the so-called "push" or "pull" BEV projection approaches [24, 46]. The former are derived from the seminal Lift-Splat-Shoot (LSS) [58] that leverages depth uncertainty estimations from each view to project features in a shared BEV space. The predictive performance of LSS can be further improved with other supervision signals, e.g., depth from Lidar points [42, 61], stereo from time [41, 44, 71] or different depth parameterization [80]. Runtime speed can be significantly reduced thanks to custom pooling strategies [33, 34, 43, 51, 73, 84]. *Pull* approaches forego depth estimation and, instead, map 3D locations from the BEV space to the image space of the cameras. Then, they collect image features with deformable attention [45] or bilinear interpolation [24] and spread them in the 3D space filling it. Pull approaches have a simpler projection and have also been largely adopted and improved [14, 38, 78]. While predictive performance of BEV methods has been improved significantly in the last two years, very few methods deal with annotation-efficient learning beyond data augmentation strategies in the image or BEV space [34, 43]. Obtaining precise 3D annotations for BEV perception is a costly multi-stage labour-intensive process involving annotation of both point clouds and im-

ages from multiple cameras. In this work we propose a strategy to improve the per-annotation efficiency of different BEV models and showcase it on two types of view-projections: SimpleBEV [24] and BEVFormer [45].

Self-supervised representation learning is a prominent paradigm that leverages unlabelled data towards producing useful representations (generalizable, robust, ready-to-use) for different tasks of interest. Self-supervised learning (SSL) defines an annotation-free *pretext task* that is determined solely by raw data, with the aim of providing supervision signal to extract useful patterns from the data and to learn representations. SSL pretrained models are subsequently used *off-the-shelf*, probed or finetuned on different tasks of interest with limited amounts of labels, displaying better performance per-annotation compared to fully supervised counterparts [28]. In the image domain, a myriad of SSL pretext tasks have been proposed such as predicting perturbations incurred into an image [19, 20, 82], contrastive learning by separating similar from dissimilar views [15, 16, 26, 55], learning by clustering [10, 11], self-distillation [2, 5, 12, 21, 23], or most recently masked-image modeling objectives, particularly suitable for Vision Transformer architectures [4, 22, 27, 56, 85]. In spite of the success on curated internet images, e.g., ImageNet, SSL pretraining of image backbones on driving data is non-trivial, due to the high redundancy and class imbalance specific to driving data [13, 76]. However, the availability of multiple sensor types on the vehicle, such as Lidar, and the emergence of vision foundation models, opens the door to different strategies to acquire 3D and/or semantic knowledge into image encoders for driving perception, e.g., unsupervised semantic segmentation [68] or detection [65]. The synchronization of surround-view cameras and Lidar is also leveraged for pretraining Lidar networks with label-free knowledge distillation from pretrained image models leading to substantial performance gains in low-label regimes [50, 52, 59, 62] and generalization [59].

BEV pretraining and distillation. Pretraining is commonly used for perception in autonomous driving (2D/3D object detection, semantic segmentation) to boost performance and compensate for the scarcity of labeled data and task difficulty, requiring semantic and 3D geometry awareness. ImageNet or depth estimation pretraining are widely used [57] for monocular perception. For BEV perception most works start from a backbone pretrained for monocular 3D object detection [70]. The shared BEV space between cameras and Lidar enables new forms of pretraining by distilling 3D reasoning skills from Lidar networks into camera-based ones in order to compensate for potential loss of geometric information in the view-projection module [18, 30, 35, 36, 47, 72]. They take the form a teacher-student architecture, with the Lidar teacher network trained in a supervised manner on 3D annotations which are however costly to acquire. First approaches to pretrain camera-based BEV networks following the SSL paradigm with an annotation-free pretext task, such as occupancy estimation [54], forecasting Lidar point clouds [81], reconstructing 3D surfaces and RGB images [79], have emerged only very recently with promising results. However they focus mostly on purely geometric cues or rendering RGB pixels which can be sub-optimal for downstream perception performance [3] and potentially hinder the existing semantic knowledge in the image encoder. We propose to distill pretrained image features from DINOv2 [56] in the voxel space such that the produced BEV features are not only geometry aware, but also semantic aware. Closest to our work, is the recent POP-3D [69] that distills CLIP features [60] into a 3D occupancy prediction model [37] towards open-vocabulary perception and not for pretraining.

3. Method

Our goal is to pretrain a camera-only BEV segmentation network in a self-supervised way. To this end, we intend to equip the learned BEV representations with the ability to encode both the 3D geometry of the scene and semantic-aware information, crucial for downstream tasks such as semantic segmentation within the BEV space. To achieve this goal, we leverage the availability of (i) aligned Lidar and image data in autonomous driving setups and (ii) a self-supervised pretrained image encoder able to extract high-level 2D features from images (e.g., DINOv2 [56]). The proposed self-supervised BEV pretraining method OccFeat, illustrated in Figure 2, encompasses two training objectives:

Occupancy reconstruction ($\mathcal{L}_{\text{occ}}$): This objective enforces the BEV network to capture the 3D geometry of the scene through an occupancy reconstruction task, defined using the available Lidar data.

Occupancy-guided feature distillation ($\mathcal{L}_{\text{feat}}$): This objective enforces the BEV network to reconstruct high-level semantic features. The network is trained to predict, at occupied voxel locations, the features of an off-the-shelf self-supervised pretrained image encoder.

The total objective that our self-supervised pre-training approach minimizes is:

$$\mathcal{L} = \mathcal{L}_{\text{occ}} + \lambda \cdot \mathcal{L}_{\text{feat}}, \tag{1}$$

where λ is the weight coefficient for balancing the two loss terms. Except otherwise stated, we use $\lambda = 0.01$.

In the following, we begin with a brief overview of camera-only BEV networks in Sec. 3.1. Then, we describe our occupancy reconstruction objective in Sec. 3.2 and our occupancy-guided feature distillation objective in Sec. 3.3.

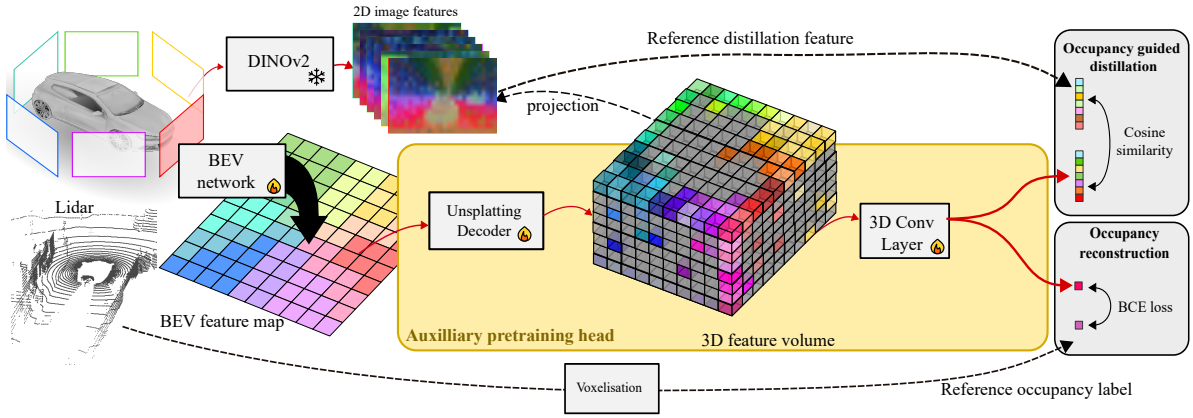


Figure 2. **Overview of OccFeat's self-supervised BEV pretraining approach.** OccFeat attaches an auxiliary pretraining head on top of the BEV network. This head "unsplats" the BEV features to a 3D feature volume and predicts with it (a) the 3D occupancy of the scene (occupancy reconstruction loss) and (b) high-level self-supervised image features characterizing the occupied voxels (occupancy-guided distillation loss). The occupancy targets are produced by "voxelizing" Lidar points (see Fig. 3), while the self-supervised image foundation model DINOv2 provides the feature targets for the occupied voxels. The pretraining head is removed after the pretraining.

3.1. BEV networks

The BEV networks aim to build a BEV feature map from registered image data. In our case, this feature map is used for semantic segmentation. These networks share a common architecture composed of 1) an image encoder, 2) a projection module and 3) a decoder.

The image encoder E_{I} produces a feature map F_c for each image I_c, with c ranging from 1 to C, where C is the total number of surround-view cameras in the vehicle. Typically, these encoders come either from ResNet [25] or EfficientNet [64] family of models.

The projection module P_{B} is the module responsible for changing the representation space, from the sensor coordinate system (image features F_c) to the BEV space. P_{B} takes as input the image features $\{F_c\}_{c=1}^{C}$ and camera calibration and projects the image features in the BEV space. Architectures differ in the way they operate this projection, from a full image feature volume aggregated over the vertical axis (SimpleBEV [24]) or a sparser volume filled according to an estimated depth distribution (LSS [58]) to an attention-based projection as in CVT [83], BEVFormer [45].

The decoder D_{B} takes as input the image features in the BEV space generated by the P_{B} and further processes them with 2D convolutional layers and optionally upsamples them to the desired segmentation map resolution [24, 58].

This produces the output BEV features $F_{\text{B}} \in \mathbb{R}^{N_{\text{B}} \times H_{\text{B}} \times W_{\text{B}}}$, where $H_{\text{B}} \times W_{\text{B}}$ is the spatial resolution of the BEV features and N_{B} is the number of feature channels.

Architecture-agnostic BEV representation pretraining. The self-supervised pretraining approach that we propose is applied on these BEV features F_{B} that D_{B} produces. Hence, it is possible to apply this pretraining approach to any BEV model by plugging in the pretraining head, which we describe next, at the end of the BEV network, before the downstream task-specific head. We note that the auxiliary pretraining head is removed after the end of pretraining.

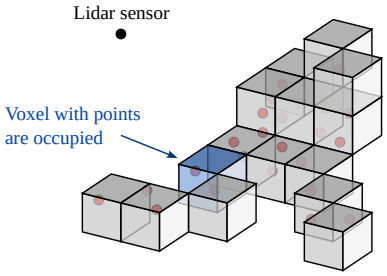


Figure 3. **Occupancy grid.** A voxel is considered occupied if there is at least one point inside it.

3.2. Occupancy reconstruction

Building on the insights from previous studies that have highlighted the effectiveness of reconstruction as a valuable prior for diverse modalities, such as images [4, 27] and Lidar point clouds [7], we employ a simple occupancy reconstruction pretraining task for BEV networks. The goal is to lead the BEV network to learn BEV features that encode information about the 3D geometry of the scene.

Let $\mathcal{V}$ represent a voxel grid with shape $Z_{\text{B}} \times H_{\text{B}} \times W_{\text{B}}$, where Z_{B} is the height dimension. To create the occupancy targets $O \in \{0,1\}^{Z_{\text{B}} \times H_{\text{B}} \times W_{\text{B}}}$, we adhere to the common convention [37, 66] wherein a voxel $v \in \mathcal{V}$ is considered occupied if at least one Lidar point falls inside (see Fig. 3).

To estimate the occupancy grid $\hat{O}$, we employ an "unsplatting" decoder network D_{V} that takes the 2D BEV feature maps $F_{\text{B}} \in \mathbb{R}^{N_{\text{B}} \times H_{\text{B}} \times W_{\text{B}}}$ from D_{B} as input and generates a 3D feature volume $F_{\text{V}} \in \mathbb{R}^{N_{\text{B}} \times Z_{\text{B}} \times H_{\text{B}} \times W_{\text{B}}}$.

The unsplatting decoder D_{V} starts with two 2D convolutional layers. The first layer, with 3×3 kernels and N_{B} output channels, is followed by Instance Normalization [67] and ReLU units. The second layer has 1×1 kernels with

$N_{\text{B}} Z_{\text{B}}$ output channels. These layers produce 2D BEV feature maps of shape $(N_{\text{B}} Z_{\text{B}}) \times H_{\text{B}} \times W_{\text{B}}$, which are reshaped into a 3D feature volume of shape $N_{\text{B}} \times Z_{\text{B}} \times H_{\text{B}} \times W_{\text{B}}$. This reshaping is done by dividing the $(N_{\text{B}} Z_{\text{B}})$-dimensional feature channels into Z_{B} groups, each with N_{B} channels.

Next, the decoder D_{V} processes these 3D features with two 3D convolutional layers. The first layer, with $1\times1\times1$ kernels and $2N_{\text{B}}$ output channels, is followed by a Softplus non-linearity. The second layer has $1\times1\times1$ kernels with N_{B} output channels, producing the final 3D feature volume $F_{\text{V}} \in \mathbb{R}^{N_{\text{B}} \times Z_{\text{B}} \times H_{\text{B}} \times W_{\text{B}}}$.

Finally, to generate the occupancy prediction $\hat{O}$, a single 3D convolution layer with a $1\times1\times1$ kernel is applied on F_{V}, followed by a sigmoid activation.

The loss function to minimize is a binary cross-entropy loss on the voxel occupation:

$$\mathcal{L}_{\text{occ}} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \text{BCE}(\hat{O}_v, O_v). \tag{2}$$

3.3. Occupancy-guided feature distillation

We introduce a self-supervised objective that complements occupancy reconstruction by guiding our BEV network to encode high-level semantic information. To achieve this, we leverage the availability of a self-supervised pretrained image network, denoted as E_{I}^{Y}, which takes an image as input and produces high-level 2D feature maps with N_{y} feature channels. Our OccFeat approach involves a feature distillation objective, where we fill the occupied voxels in $\mathcal{V}$ with features extracted from E_{I}^{Y} and then train the BEV network to predict these voxel features.

Let $\mathcal{V}_{\text{Occ}}$ represent the set of occupied voxels, defined as $\mathcal{V}_{\text{Occ}} = \{\forall v \in \mathcal{V} \,|\, O_v = 1\}$. Our feature distillation objective operates specifically on these occupied voxels. To create the target feature $Y_v \in \mathbb{R}^{N_{\text{y}}}$ for each occupied voxel $v \in \mathcal{V}_{\text{Occ}}$, we project the voxel's center 3D coordinates onto the image feature maps extracted by the target image encoder E_{I}^{Y} from the surround-view input images $\{I_c\}_{c=1}^{C}$. Given these projections of 3D points into 2D images, we obtain an N_y-dimensional feature vector by bilinearly sampling a feature map of each image I_c with a valid projection (i.e., if a point is projected inside an image I_c). The target feature Y_v is then computed across images with valid projections as the average of the bilinearly-sampled feature vectors. To predict the target features of the occupied voxels, we use the 3D feature volume $F_{\text{V}} \in \mathbb{R}^{N_{\text{B}} \times Z_{\text{B}} \times H_{\text{B}} \times W_{\text{B}}}$ produced by the D_{V} decoder. A single 3D convolution layer with $1\times1\times1$ kernels and N_{y} output channels is applied on F_{V}, resulting in the 3D feature volume $\hat{Y} \in \mathbb{R}^{N_{\text{y}} \times Z_{\text{B}} \times H_{\text{B}} \times W_{\text{B}}}$. Then, for each occupied voxel $v \in \mathcal{V}_{\text{Occ}}$, we extract its corresponding feature $\hat{Y}_v \in \mathbb{R}^{N_{\text{y}}}$ from $\hat{Y}$.

The feature distillation loss that we aim to minimize is the average negative cosine similarity between the predicted and target features for each occupied voxel $v \in \mathcal{V}_{\text{Occ}}$:

$$\mathcal{L}_{\text{feat}} = -\frac{1}{|\mathcal{V}_{\text{Occ}}|} \sum_{v \in \mathcal{V}_{\text{Occ}}} \cos(\hat{Y}_v, Y_v). \tag{3}$$

We note that there exists a small number of occupied voxels that lack valid projections into any of the images $\{I_c\}_{c=1}^{C}$. Although not explicitly shown in Eq. (3) for the sake of notation simplicity, these voxels are, in fact, excluded from the computation of the feature distillation loss.

4. Experiments

4.1. Experimental setup

Datasets. To evaluate our approach we use the nuScenes [9] dataset for both conducting the pretraining of the camera-only BEV networks and for finetuning them on the downstream tasks of BEV-based semantic segmentation. The dataset is composed of 1,000 sequences recorded in Boston and Singapore. The data is divided in training (700 sequences), validation (100 sequences) and test (200 sequences) splits. Each frame contains a point cloud acquired with 32-layer Lidar and 6 images covering the surroundings of the ego-vehicle .

BEV segmentation tasks. We consider two different tasks related to BEV segmentation. First, as in [24], we evaluate our method on vehicle segmentation. We build target BEV segmentation ground truth by projecting the boxes of vehicles on the BEV plane. We use the same setting as [24], i.e., a range of 50 meters around the ego-vehicle, and BEV ground truths of size 200x200 pixels. Second, we focus on the layout and evaluate on map segmentation. Here, we evaluate the ability of our finetuned BEV network to segment bakcground classes: "road", "sidewalk", "crosswalk", "parking area" and "road dividers".

Architectures. *BEV networks*. We experiment with two BEV architectures: SimpleBEV [24] and BEVFormer [45] (implementation from [24]).

Image encoders. As image backbones in the BEV segmentation networks, we employ either EfficientNet-B0 [64] (EN-B0) or ResNet-50 [25] (RN-50). Following the common practice in camera-only BEV segmentation [24], these image backbones have undergone pretraining on ImageNet, either in a supervised way (for EN-B0) or self-supervised using MoCov2 [17] (for RN-50).

Teacher model. The self-supervised image foundation model E_{I}^{Y}, used as the teacher for the occupancy-guided feature distillation, is the ViT-S/14 variant of DINOv2 [56].

OccFeat pretraining. We pretrain BEV segmentation networks with batch size 16 on 4×V100 GPUs using the Adam optimizer with weight decay $1e{-}7$ and a constant learning rate of $1e{-}3$ during training. By default we pretrain for 50 epochs. We also pretrain some models for 100 epochs,

Architecture	Image Backb.	BEV Pretraining	Vehicles 1%	Vehicles 10%	Vehicles 100%	Map 1%	Map 10%	Map 100%
SimpleBEV	EN-B0	None	13.7	26.0	37.4	20.1	28.1	45.3
		Img-ALSO	18.5 (+4.8)	26.6 (−0.6)	32.1 (−5.3)	23.2 (+3.1)	29.0 (+0.9)	41.1 (−4.2)
		OccFeat	24.3 (+10.6)	30.9 (+4.9)	37.7 (+0.3)	25.4 (+5.3)	33.7 (+5.6)	47.3 (+2.0)
		OccFeat†	24.5 (+10.8)	32.0 (+6.0)	38.1 (+0.7)	26.2 (+6.1)	34.3 (+6.2)	47.6 (+2.3)
	RN-50	None	13.9	28.6	42.2	20.1	31.2	44.2
		Img-ALSO	18.4 (+4.5)	25.6 (−3.0)	37.2 (−5.0)	22.3 (+2.2)	28.7 (−2.5)	41.2 (−3.0)
		OccFeat	23.5 (+9.6)	31.4 (+2.8)	41.0 (−1.2)	25.2 (+5.1)	33.6 (+2.4)	50.0 (+5.8)
		OccFeat†	24.8 (+10.9)	31.3 (+2.7)	41.7 (−0.5)	25.9 (+5.8)	34.4 (+3.2)	51.3 (+7.1)
BEVFormer	EN-B0	None	11.3	22.8	37.2	20.4	29.4	49.6
		OccFeat	24.9 (+13.6)	30.1 (+7.3)	38.8 (+1.6)	26.6 (+6.2)	36.2 (+6.8)	50.5 (+0.9)

Table 1. **Segmentation IoU results for Vehicle and Map classes**. Comparing our OccFeat against no BEV pretraining (None) and the pretraining baseline Img-ALSO. Results with 224×400 image resolution. †: pretrained for 100 epochs; all other models for 50 epochs.

which we denote with OccFeat† in the tables. We use augmentations from [24]. Except otherwise stated, we use 224×400 image resolution for the input camera frames.

Finetuning. We experiment with finetuning at 1%, 10% and 100%. We use the the AdamW optimizer and One Cycle scheduler with $1e{-}7$ as weight decay. The complete hyperparameters (number of epochs / starting learning rate / batch size / batch gradient accumulation) are the following, 1%: 100/$1e{-}4$/6/1; 10%: 50/$1e{-}4$/6/1 and 100%: 30/$3e{-}4$/8/5. The training is conducted with 2×V100 GPUs. If the BEV segmentation network undergoing finetuning has not been pretrained with OccFeat, we initialize its image backbone with weights pretrained on ImageNet, following the same initialization procedure used before the OccFeat pretraining. This is common practice in camera-only BEV segmentation [24].

4.2. Pretraining baselines

To validate the effectiveness of our OccFeat approach, which integrates feature distillation with 3D geometry prediction, we implement a pretraining baseline focusing solely on 3D geometry reconstruction. To implement this baseline, called Img-ALSO, we modify the Lidar pretraining method ALSO [7] for the purpose of pretraining camera-only BEV networks. ALSO is a top-performing reconstruction-based approach for self-supervised pretraining Lidar networks. Its pretext task is to learn an implicit function partitioning the space between empty and occupied (inside objects) spaces, where the supervision signal is directly generated from the input Lidar data, using the sensor line of sights. This entails a geometry reconstruction task, where each pair of 3D point/output feature must reconstruct a local neighborhood. To implement our Img-ALSO baseline, we begin with the ALSO variant tailored for pretraining Lidar detection networks, such as the SECOND network [77], where the features from the input 3D point cloud are projected on a top-view plane. In this setup, we replace the 3D backbone with the image BEV network intended for pretraining. As the supervision signal in Img-ALSO still comes from the Lidar points (as in ALSO), we use the same hyper-parameters as ALSO, i.e., decimation grid of 10cm, maximal distance-to-Lidar-point for query of 10cm and a reconstruction radius of 1m. We compare against Img-ALSO in Sec. 4.3.

In addition to comparing OccFeat with the above baseline, in Sec. 4.4 we include comparisons against ablations that use only the occupancy reconstruction objective ($\mathcal{L}_{\text{occ}}$), which is another 3D geometry-based approach, or only the occupancy-guided feature distillation objective ($\mathcal{L}_{\text{feat}}$).

4.3. Results

Comparison with baselines. In the SimpleBEV results presented in Tab. 1, our OccFeat self-supervised BEV pretraining approach is compared against the proposed baseline Img-ALSO, along with the case of not conducting BEV pretraining. We provide results for both Vehicle and Map segmentation, utilizing 1%, 10%, and 100% of annotated training data. For the SimpleBEV network we use either the EN-B0 or the RN-50 image backbones and 224×400 image resolution for the input camera frames.

We observe that, compared to the absence of BEV pretraining, our OccFeat enhances segmentation results in almost all settings. The only exception is that of Vehicle segmentation with 100% annotations using SimpleBEV with RN-50. The improvement is particularly prominent with only 1% or 10% of annotated data, showcasing the effectiveness of our approach in low-shot settings. When comparing with the examined baseline (Img-ALSO), our OccFeat outperforms them in almost all cases. An interesting observation is that Img-ALSO improves segmentation performance only in the 1%-annotation settings, while yielding worse results for the 10%- and 100%-annotation settings.

Scaling with longer BEV pretraining. In Tab. 1, apart from presenting results for 50 pretraining epochs, we also

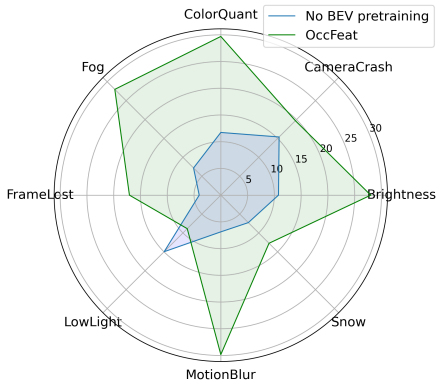


Figure 4. **Study on robustness**. Segmentation results on nuScenes-C dataset for Vehicle classes using BEVFormer network with EN-B0 image backbone on 100% annotated data. Comparison of our OccFeat against no BEV pretraining.

BEV Pretraining	Image Resolution		Vehicles		
	Pretraining	Finetuning	1%	10%	100%
None	N/A	224×400	13.7	26.0	37.4
OccFeat	224×400		24.3	30.9	37.7
OccFeat†	224×400		24.5	32.0	38.1
None	N/A	448×800	12.9	28.4	41.6
OccFeat	448×800		26.5	34.5	41.5
OccFeat†	224×400		26.1	33.3	41.5

Table 2. **Impact of image resolution**. SimpleBEV segmentation IoU results for the Vehicle class. Results with the EN-B0 image backbone. †: pretrained for 100 epochs.

include results for our OccFeat approach and SimpleBEV (with EN-B0 or RN-50) with an extended pretraining duration of 100 epochs (OccFeat† model). We observe consistent improvements in segmentation results with this longer pretraining duration. This observation underscores the scalability of our approach, demonstrating its ability to gain further benefits from longer pretraining periods.

Adaptable to various BEV network architectures. As mentioned in Sec. 3, our BEV pretraining method can be used with any BEV network architecture. In Tab. 1, apart from results with SimpleBEV networks, we also present results using the BEVFormer [45] segmentation network for the 1%, 10%, and 100% annotation settings. We compare the performance of our OccFeat method against the scenario where BEV pretraining is not conducted. We see that even when employing the BEVFormer network architecture, our OccFeat pretraining approach enhances segmentation performance across all evaluation settings.

Exploiting higher resolution images. In Tab. 2, we present segmentation results using either 224×400 or 448×800 resolutions for the camera frames fed into the SimpleBEV network. Our OccFeat pretraining approach consistently enhances segmentation results across all cases. What is noteworthy is that pretraining with the lower 224×400 resolution and then fine-tuning with the higher 448×800 resolution also improves the results, almost as much as pretraining directly with the higher resolution.

BEV Network	BEV Pretraining	OccFeat losses		Vehicles	Maps
		$\mathcal{L}_{occ}$	$\mathcal{L}_{feat}$	1%	1%
SimpleBEV	✗			13.7	20.1
	✓	✓		22.8	25.2
	✓		✓	17.4	23.4
	✓	✓	✓	24.3	25.4
BEVFormer	✗			11.3	20.4
	✓	✓		21.9	25.1
	✓	✓	✓	24.9	26.6

Table 3. **Ablation of OccFeat's losses**. BEV segmentation results (IoU) for the Vehicle and Map classes using 1% annotated data. Results with 224×400 resolution and the EN-B0 image backbone.

Loss weight λ	0	0.0001	0.001	0.01	0.1	1.0
Vehicle (1%)	22.8	22.9	22.7	**24.3**	23.9	19.4

Table 4. **Impact of loss weight** λ ($\mathcal{L} = \mathcal{L}_{occ} + \lambda \cdot \mathcal{L}_{feat}$). SimpleBEV vehicle segmentation results using 1% annotated data, 224×400 image resolution and the EN-B0 image backbone.

Robustness study. We study the robustness of OccFeat by evaluating it on the nuScenes-C benchmark [74]. This benchmark consists of eight distinct data corruptions, each with three intensity levels, applied to the validation set of nuScenes. In Fig. 4 we present vehicle segmentation results on nuScenes-C using BEVFormer with EN-B0 backbone finetuned on 100% annotation data. For each corruption type we report the average across three severity levels. The comparison of our OccFeat against no BEV pretraining illustrates that the OccFeat pretraining improves the robustness of the final BEV model.

4.4. Ablation study

Ablation of OccFeat's losses. In Tab. 3 we conduct an ablation study on the two pretraining objectives, namely $\mathcal{L}_{occ}$ and $\mathcal{L}_{feat}$, of our OccFeat approach. The evaluation focuses on Vehicle and Map segmentation results with 1% annotated training data, using both the SimpleBEV and BEVFormer networks. For SimpleBEV, both $\mathcal{L}_{occ}$ (a 3D geometry prediction objective) and $\mathcal{L}_{feat}$ (an occupancy-guided feature distillation objective) showcase improvements compared to the scenario without BEV pretraining. The combination of both pretraining objectives, forming our OccFeat approach, yields the most favorable segmentation results. This underscores the efficacy of our OccFeat pretraining method, distinguishing it from prior self-supervised BEV pretraining works that rely solely on 3D geometry prediction. The advantage of enhancing 3D geometry prediction ($\mathcal{L}_{occ}$) with feature distillation ($\mathcal{L}_{feat}$) is further affirmed by the ablation results obtained with BEVFormer.

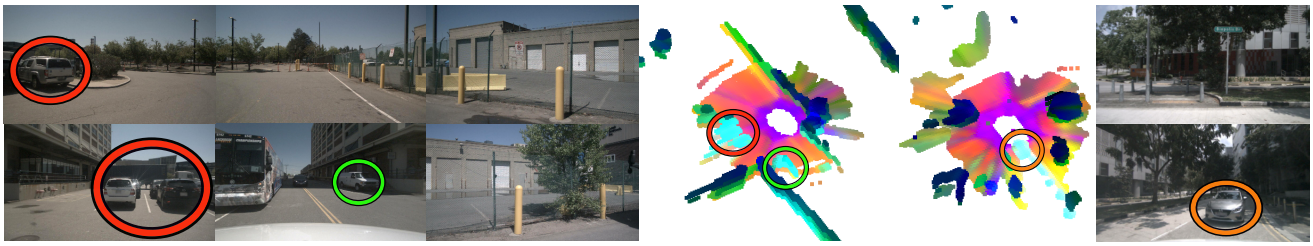
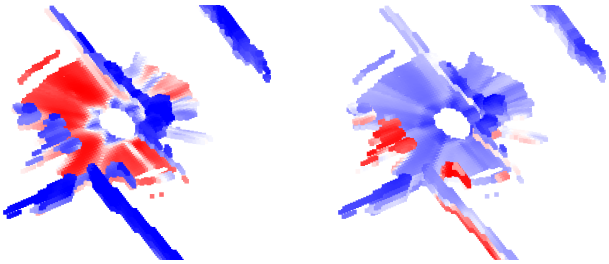


Figure 5. **Visualisation of predicted 3D features,** using a 3-dimensional PCA mapped on RGB channels. The features contain semantic information, e.g., cars in cyan color. Using the same PCA mapping on a different scene (right), we show that semantic features are consistent across scenes. Objects within colored circles in the feature space correspond to those in similarly colored circles in the image.



(a) Road (b) Car

Figure 6. **Correlation maps** of the student's predicted 3D features and features selected on the road (a) and on a car (b).

Impact of loss weight λ. In Tab. 4, we investigate the influence of the loss weight λ, responsible for balancing the two loss terms of OccFeat (refer to Eq. (1)). The most favorable segmentation results are achieved for λ values ranging between 0.001 and 0.01, with 0.01 being the optimal choice.

4.5. Qualitative results

In Fig. 5 and Fig. 6, we assess the semantic quality of the reconstructed features, using a colored mapping from PCA (Fig. 5) and correlation maps (Fig. 6). We can observe that semantic information from DINOv2 teacher has been preserved and semantic classes such as cars are easily separable. Additionally, the representations are consistent accross scenes, e.g., on Fig. 5 (right), we apply the PCA mapping computed on the left scene to a new scene. As an example, the cyan points on the right correspond to a car.

5. Conclusion

We introduced OccFeat, a self-supervised pretraining method for camera-only BEV segmentation networks. Our approach combines two pretraining objectives: a 3D occupancy prediction task using raw Lidar data and an occupancy-guided feature distillation task based on the self-supervised pre-trained image foundation model DINOv2. The former enhances the learning of 3D geometry-aware BEV features, while the latter focuses on semantic-aware BEV features. Our empirical results demonstrate that both pretraining objectives enhance segmentation performance compared to not conducting BEV pre-training. Combining both objectives yields the most favorable results, emphasizing the effectiveness of our OccFeat approach. This sets our method apart from prior self-supervised BEV pre-training methods that solely rely on 3D geometry prediction.

Limitations and Perspectives. While OccFeat has proven highly effective for low-data scenarios (i.e., 1% and 10% finetuning), it yields slight or no improvements in the 100% finetuning setting. This could be because we self-supervisedly pretrained and then supervisedly finetuned on the same data, leaving all information available at finetuning. Additionally, the nuScenes dataset used for pretraining is relatively small. Perhaps *using a larger pretraining dataset* could further enhance performance in the 100% finetuning setting.

One aspect not sufficiently explored in our work is *"scaling the pretraining"*. As demonstrated in ScaLR [59], when pretraining Lidar networks via self-supervised distillation of image-based models, scaling the teacher and student can significantly boost performance. For example, using a larger teacher model, such as going from ViT-S to ViT-L or ViT-G variants of DINOv2, could yield superior features for OccFeat's distillation loss. Similarly, scaling the student components —specifically, the image encoder and BEV decoder—might enhance OccFeat's distillation process.

Furthermore, another avenue for improvement could involve *incorporating time* into our self-supervised pretraining task. For instance, accumulating Lidar points over several frames to generate denser occupancy maps could enhance the effectiveness of OccFeat's pretraining.

Acknowledgements This work was performed using HPC resources from GENCI–IDRIS (Grants 2022-AD011012883R2, 2022-AD011012884R2 and 2024-AD011015037). It was supported in part by the French Agence Nationale de la Recherche (ANR) grant MultiTrans (ANR21-CE23-0032). It also received the support of CTU Student Grant SGS21184OHK33T37 and by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90254), and the support of EXA4MIND, a European Union's Horizon Europe Research and Innovation programme under grant agreement N°101092944. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them. The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Adil Kaan Akan and Fatma Güney. Stretchbev: Stretching future instance prediction spatially and temporally. In *ECCV*, 2022. 1

[2] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *CVPR*, 2023. 3

[3] Randall Balestriero and Yann LeCun. Learning by reconstruction produces uninformative features for perception. *arXiv preprint arXiv:2402.11337*, 2024. 3

[4] Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT pre-training of image transformers. In *ICLR*, 2022. 1, 3, 4

[5] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022. 3

[6] Florent Bartoccioni, Éloi Zablocki, Andrei Bursuc, Patrick Pérez, Matthieu Cord, and Karteek Alahari. Lara: Latents and rays for multi-camera bird's-eye-view semantic segmentation. In *CoRL*, 2022. 1, 2

[7] Alexandre Boulch, Corentin Sautier, Björn Michele, Gilles Puy, and Renaud Marlet. Also: Automotive lidar self-supervision by occupancy estimation. In *CVPR*, 2023. 1, 4, 6

[8] Thibault Buhet, Emilie Wirbel, Andrei Bursuc, and Xavier Perrotton. Plop: Probabilistic polynomial objects trajectory planning for autonomous driving. In *CoRL*, 2020. 1

[9] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 2, 5

[10] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 3

[11] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 3

[12] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 1, 3

[13] Kai Chen, Lanqing Hong, Hang Xu, Zhenguo Li, and Dit-Yan Yeung. Multisiam: Self-supervised multi-instance siamese representation learning for autonomous driving. In *ICCV*, 2021. 3

[14] Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Chang Huang, and Wenyu Liu. Polar parametrization for vision-based surround-view 3d detection. *arXiv preprint arXiv:2206.10965*, 2022. 2

[15] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1, 3

[16] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 3

[17] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 5

[18] Zehui Chen, Zhenyu Li, Shiquan Zhang, Liangji Fang, Qinhong Jiang, and Feng Zhao. Bevdistill: Cross-modal bev distillation for multi-view 3d object detection. In *ICLR*, 2023. 3

[19] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *CVPR*, 2015. 3

[20] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 3

[21] Spyros Gidaris, Andrei Bursuc, Gilles Puy, Nikos Komodakis, Matthieu Cord, and Patrick Pérez. Obow: Online bag-of-visual-words generation for self-supervised learning. In *CVPR*, 2021. 1, 3

[22] Spyros Gidaris, Andrei Bursuc, Oriane Simeoni, Antonin Vobecky, Nikos Komodakis, Matthieu Cord, and Patrick Pérez. Moca: Self-supervised representation learning by predicting masked online codebook assignments. *TMLR*, 2024. 3

[23] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 2020. 1, 3

[24] Adam W Harley, Zhaoyuan Fang, Jie Li, Rares Ambrus, and Katerina Fragkiadaki. Simple-bev: What really matters for multi-sensor bev perception? In *ICRA*, 2023. 1, 2, 3, 4, 5, 6

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4, 5

[26] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 3

[27] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1, 3, 4

[28] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *ICML*, 2020. 3

[29] Noureldin Hendy, Cooper Sloan, Feng Tian, Pengfei Duan, Nick Charchut, Yuesong Xie, Chuang Wang, and James Philbin. Fishing net: Future inference of semantic heatmaps in grids. In *CVPR*, 2020. 1

[30] Yu Hong, Hang Dai, and Yong Ding. Cross-modality knowledge distillation network for monocular 3d object detection. In *ECCV*, 2022. 3

[31] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In *ICCV*, 2021. 1

[32] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *CVPR*, 2023. 1

[33] Junjie Huang and Guan Huang. Bevpoolv2: A cutting-edge implementation of bevdet toward deployment. *arXiv preprint arXiv:2211.17111*, 2022. 2

[34] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 1, 2

[35] Linyan Huang, Huijie Wang, Jia Zeng, Shengchuan Zhang, Liujuan Cao, Rongrong Ji, Junchi Yan, and Hongyang Li. Geometric-aware pretraining for vision-centric 3d object detection. *arXiv preprint arXiv:2304.03105*, 2023. 3

[36] Peixiang Huang, Li Liu, Renrui Zhang, Song Zhang, Xinli Xu, Baichao Wang, and Guoyi Liu. Tig-bev: Multi-view bev 3d object detection via target inner-geometry learning. *arXiv preprint arXiv:2212.13979*, 2022. 3

[37] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *CVPR*, 2023. 3, 4

[38] Yanqin Jiang, Li Zhang, Zhenwei Miao, Xiatian Zhu, Jin Gao, Weiming Hu, and Yu-Gang Jiang. Polarformer: Multi-camera 3d object detection with polar transformer. In *AAAI*, 2023. 2

[39] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *ICRA*, 2022. 1

[40] Siyu Li, Kailun Yang, Hao Shi, Jiaming Zhang, Jiacheng Lin, Zhifeng Teng, and Zhiyong Li. Bi-mapper: Holistic bev semantic mapping for autonomous driving. *RA-L*, 2023. 1

[41] Yinhao Li, Han Bao, Zheng Ge, Jinrong Yang, Jianjian Sun, and Zeming Li. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. In *AAAI*, 2023. 2

[42] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *AAAI*, 2023. 2

[43] Yangguang Li, Bin Huang, Zeren Chen, Yufeng Cui, Feng Liang, Mingzhu Shen, Fenggang Liu, Enze Xie, Lu Sheng, Wanli Ouyang, et al. Fast-bev: A fast and strong bird's-eye view perception baseline. *arXiv preprint arXiv:2301.12511*, 2023. 2

[44] Yinhao Li, Jinrong Yang, Jianjian Sun, Han Bao, Zheng Ge, and Li Xiao. Bevstereo++: Accurate depth estimation in multi-view 3d object detection via dynamic temporal stereo. *arXiv preprint arXiv:2304.04185*, 2023. 2

[45] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, 2022. 1, 2, 3, 4, 5, 7

[46] Zhiqi Li, Zhiding Yu, Wenhai Wang, Anima Anandkumar, Tong Lu, and Jose M Alvarez. Fb-bev: Bev representation from forward-backward view transformations. In *ICCV*, 2023. 2

[47] Jihao Liu, Tai Wang, Boxiao Liu, Qihang Zhang, Yu Liu, and Hongsheng Li. Geomim: Towards better 3d knowledge transfer via masked image modeling for multi-view 3d understanding. In *CVPR*, 2023. 3

[48] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *ECCV*, 2022. 2

[49] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Aqi Gao, Tiancai Wang, and Xiangyu Zhang. Petrv2: A unified framework for 3d perception from multi-camera images. In *ICCV*, 2023. 2

[50] Youquan Liu, Lingdong Kong, Jun Cen, Runnan Chen, Wenwei Zhang, Liang Pan, Kai Chen, and Ziwei Liu. Segment any point cloud sequences by distilling vision foundation models. In *NeurIPS*, 2024. 3

[51] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *ICRA*, 2023. 1, 2

[52] Anas Mahmoud, Jordan SK Hu, Tianshu Kuai, Ali Harakeh, Liam Paull, and Steven L Waslander. Self-supervised image-to-point distillation via semantically tolerant contrastive loss. In *CVPR*, 2023. 3

[53] Yunze Man, Liang-Yan Gui, and Yu-Xiong Wang. Bev-guided multi-modality fusion for driving perception. In *CVPR*, 2023. 1

[54] Chen Min, Liang Xiao, Dawei Zhao, Yiming Nie, and Bin Dai. Uniscene: Multi-camera unified pre-training via 3d scene reconstruction. *RA-L*, 2024. 2, 3

[55] Ishan Misra and Laurens Van Der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020. 3

[56] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *TMLR*, 2023. 2, 3, 5

[57] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *ICCV*, 2021. 3

[58] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 2020. 1, 2, 4

[59] Gilles Puy, Spyros Gidaris, Alexandre Boulch, Oriane Siméoni, Corentin Sautier, Patrick Pérez, Andrei Bursuc, and Renaud Marlet. Three pillars improving vision foundation model distillation for lidar. In *CVPR*, 2024. 3, 8

[60] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 3

[61] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *CVPR*, 2021. 2

[62] Corentin Sautier, Gilles Puy, Spyros Gidaris, Alexandre Boulch, Andrei Bursuc, and Renaud Marlet. Image-to-lidar self-supervised distillation for autonomous driving data. In *CVPR*, 2022. 3

[63] Sunando Sengupta, Paul Sturgess, L'ubor Ladickỳ, and Philip HS Torr. Automatic dense visual semantic mapping from street-level imagery. In *IROS*, 2012. 2

[64] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 4, 5

[65] Hao Tian, Yuntao Chen, Jifeng Dai, Zhaoxiang Zhang, and Xizhou Zhu. Unsupervised object detection with lidar clues. In *CVPR*, 2021. 3

[66] Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, et al. Scene as occupancy. In *ICCV*, 2023. 4

[67] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 4

[68] Antonin Vobecky, David Hurych, Oriane Siméoni, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, and Josef Sivic. Drive&segment: Unsupervised semantic segmentation of urban scenes via cross-modal distillation. In *ECCV*, 2022. 3

[69] Antonin Vobecky, Oriane Siméoni, David Hurych, Spyridon Gidaris, Andrei Bursuc, Patrick Pérez, and Josef Sivic. Pop-3d: Open-vocabulary 3d occupancy prediction from images. In *NeurIPS*, 2023. 3

[70] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *ICCV*, 2021. 3

[71] Zengran Wang, Chen Min, Zheng Ge, Yinhao Li, Zeming Li, Hongyu Yang, and Di Huang. Sts: Surround-view temporal stereo for multi-view 3d detection. *arXiv preprint arXiv:2208.10145*, 2022. 2

[72] Zeyu Wang, Dingwen Li, Chenxu Luo, Cihang Xie, and Xiaodong Yang. Distillbev: Boosting multi-camera 3d object detection with cross-modal knowledge distillation. In *CVPR*, 2023. 3

[73] Enze Xie, Zhiding Yu, Daquan Zhou, Jonah Philion, Anima Anandkumar, Sanja Fidler, Ping Luo, and Jose M Alvarez. M^2bev: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation. *arXiv preprint arXiv:2204.05088*, 2022. 2

[74] Shaoyuan Xie, Lingdong Kong, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Robobev: Towards robust bird's eye view perception under corruptions, 2023. 2, 7

[75] Kaixin Xiong, Shi Gong, Xiaoqing Ye, Xiao Tan, Ji Wan, Errui Ding, Jingdong Wang, and Xiang Bai. Cape: Camera view position embedding for multi-view 3d object detection. In *CVPR*, 2023. 2

[76] Yuwen Xiong, Mengye Ren, Wenyuan Zeng, and Raquel Urtasun. Self-supervised representation learning from flow equivariance. In *ICCV*, 2021. 3

[77] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 2018. 6

[78] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *CVPR*, 2023. 2

[79] Honghui Yang, Sha Zhang, Di Huang, Xiaoyang Wu, Haoyi Zhu, Tong He, Shixiang Tang, Hengshuang Zhao, Qibo Qiu, Binbin Lin, et al. Unipad: A universal pre-training paradigm for autonomous driving. In *CVPR*, 2024. 2, 3

[80] Jiayu Yang, Enze Xie, Miaomiao Liu, and Jose M Alvarez. Parametric depth based feature representation learning for object detection and segmentation in bird's-eye view. In *ICCV*, 2023. 2

[81] Zetong Yang, Li Chen, Yanan Sun, and Hongyang Li. Visual point cloud forecasting enables scalable autonomous driving. In *CVPR*, 2024. 2, 3

[82] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 3

[83] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *CVPR*, 2022. 1, 2, 4

[84] Hongyu Zhou, Zheng Ge, Zeming Li, and Xiangyu Zhang. Matrixvt: Efficient multi-camera to bev transformation for 3d perception. In *ICCV*, 2023. 2

[85] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. In *ICLR*, 2022. 3