LetsMap: Unsupervised Representation Learning for Label-Efficient Semantic BEV Mapping

Nikhil Gosala[1], Kürsat Petek[1], B Ravi Kiran[2], Senthil Yogamani[3], Paulo Drews-Jr[4], Wolfram Burgard[5], and Abhinav Valada[1]

[1] University of Freiburg, Germany
[2] Qualcomm SARL France
[3] QT Technologies Ireland Limited
[4] Federal University of Rio Grande, Brazil
[5] University of Technology Nuremberg, Germany
http://letsmap.cs.uni-freiburg.de

Abstract. Semantic Bird's Eye View (BEV) maps offer a rich representation with strong occlusion reasoning for various decision making tasks in autonomous driving. However, most BEV mapping approaches employ a fully supervised learning paradigm that relies on large amounts of human-annotated BEV ground truth data. In this work, we address this limitation by proposing the first unsupervised representation learning approach to generate semantic BEV maps from a monocular frontal view (FV) image in a label-efficient manner. Our approach pretrains the network to independently reason about scene geometry and scene semantics using two disjoint neural pathways in an unsupervised manner and then finetunes it for the task of semantic BEV mapping using only a small fraction of labels in the BEV. We achieve label-free pretraining by exploiting spatial and temporal consistency of FV images to learn scene geometry while relying on a novel temporal masked autoencoder formulation to encode the scene representation. Extensive evaluations on the KITTI-360 and nuScenes datasets demonstrate that our approach performs on par with the existing state-of-the-art approaches while using only 1% of BEV labels and no additional labeled data.

Keywords: Unsupervised Representation Learning · Semantic BEV Mapping · Scene Understanding

1 Introduction

Semantic Bird's Eye View (BEV) maps are essential for autonomous driving as they offer rich, occlusion-aware information for height-agnostic applications including object tracking, collision avoidance, and motion control. Instantaneous BEV map estimation that does not rely on large amounts of annotated data is crucial for the rapid deployment of autonomous vehicles in novel domains. However, the majority of existing BEV mapping approaches follow a fully supervised learning paradigm and thus rely on large amounts of annotated data in BEV, which is extremely arduous to obtain and hinders the scalability of autonomous

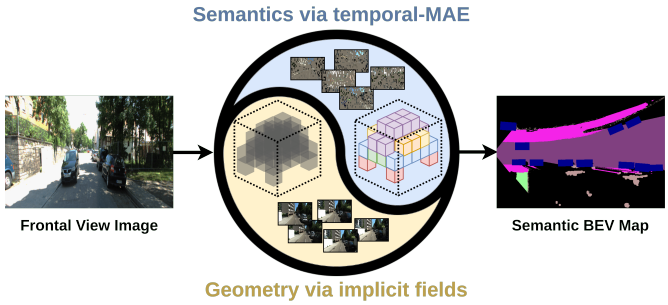


Fig. 1: LetsMap: The first unsupervised framework for label-efficient semantic BEV mapping. We use RGB image sequences to independently learn scene geometry (yellow) and scene representation (blue) in an unsupervised pretraining step, before adapting it to semantic BEV mapping in a label-efficient finetuning step.

vehicles to novel environments [23, 28–30]. Recent works circumvent this problem by leveraging frontal view (FV) semantic labels for learning both scene geometry and generating BEV pseudolabels [9], or by leveraging semi-supervised learning using pairs of labeled and unlabeled samples [7]. However, the reliance on FV labels as well as the integrated network design of both approaches gives rise to three main challenges: (1) FV labels offer scene geometry supervision only along class boundaries which limits the geometric reasoning ability of the model; (2) FV labels are dataset-specific and any change in class definition mandates full model retraining; and (3) tightly coupled network designs hinder the quick adoption of latest advances from literature.

In this work, we address these limitations by proposing the first unsupervised representation learning framework for predicting semantic BEV maps from monocular FV images in a label-efficient manner. Our approach, LetsMap, utilizes the spatiotemporal consistency and dense representation offered by FV image sequences to alleviate the need for manually annotated data. To this end, we disentangle the two sub-tasks of semantic BEV mapping, i.e., *scene geometry modeling* and *scene representation learning*, into two disjoint neural pathways (Fig. 1) and learn them using an unsupervised pretraining step. We then finetune the resultant model for semantic BEV mapping using only a small fraction of labels in BEV. LetsMap explicitly learns to model the scene geometry via the geometric pathway by leveraging implicit fields, while learning scene representations via the semantic pathway using a novel temporal masked autoencoder (T-MAE) mechanism. During pretraining, we supervise the geometric pathway by exploiting the spatial and temporal consistency of the multi-camera FV images across multiple timesteps and train the semantic pathway by enforcing reconstruction of the FV images for both the current and future timesteps using the masked image of only the current timestep. We extensively evaluate LetsMap on the KITTI-360 [21] and nuScenes [2] datasets and demonstrate that our approach performs on par with existing fully-supervised and self-supervised approaches while using only 1% of BEV labels, without leveraging any additional labeled data.

2 Related Work

In this section, we outline current work on semantic BEV mapping, monocular scene geometry estimation, and image-based scene representation learning.

BEV Segmentation: Monocular semantic BEV mapping methods typically focus on learning a lifting mechanism to transform features from FV to BEV. Early works of VED [23] and VPN [28] learn the transformation without using scene geometry, which limits their performance in the real world. PON [30] solves this issue by incorporating scene geometry into the network design while LSS [29] learns a depth distribution to transform features from FV to BEV. PanopticBEV [10] splits the world into *flat* and *non-flat* regions and transforms them to BEV using two disjoint pathways. Recent methods use transformers to generate BEV features from both single image [31] and multi-view images [37]. Some works also use multi-modal data to augment monocular cameras [11, 20, 22, 32]. All the aforementioned approaches follow a fully supervised learning paradigm and rely on vast amounts of resource-intensive human-annotated semantic BEV labels. Recent works reduce reliance on BEV ground truth labels by combining labeled and unlabeled images in a semi-supervised manner [7] or by leveraging FV labels to generate BEV pseudolabels and train the network in a self-supervised manner [9]. However, these approaches rely on additional labeled data or use tightly coupled network designs which limits their ability to scale to new environments or incorporate the latest advances in literature. In this paper, we propose a novel unsupervised label-efficient approach that first learns scene geometry and scene representation in a modular, label-free manner before adapting to semantic BEV mapping using only a small fraction of BEV semantic labels.

Monocular Scene Geometry Estimation: Scene geometry estimation is a fundamental challenge in computer vision and is a core component of 3D scene reconstruction. Initial works use techniques such as multi-view stereo [6] and visual SLAM [1, 35] while recent approaches leverage learnable functions in the form of ray distance functions [18] or implicit neural fields [25]. Early neural radiance fields-based approaches were optimized on single scenes and relied on substantial amounts of training data [25]. PixelNeRF [38] addresses these issues by conditioning NeRF on input images, enabling simultaneous optimization across different scenes. Recent works improve upon PixelNeRF by decoupling color from scene density estimation [36], and by using a tri-planar representation to query the neural field from any world point [17]. In our approach, we leverage implicit fields to generate the volumetric density from a single monocular FV image to constrain features from the uniformly-lifted 2D scene representation features.

Scene Representation Learning: Early works used augmentations such as image permutation [26], rotation prediction [8], noise discrimination [14], and frame ordering [19] to learn scene representation; which were primitive and lacked generalization across diverse tasks. [5, 13] propose using contrastive learning to learn scene representation, and [3] builds upon this paradigm by removing the need for negative samples during training. Recent works propose masked autoencoders [12] wherein masked input image patches are predicted by the

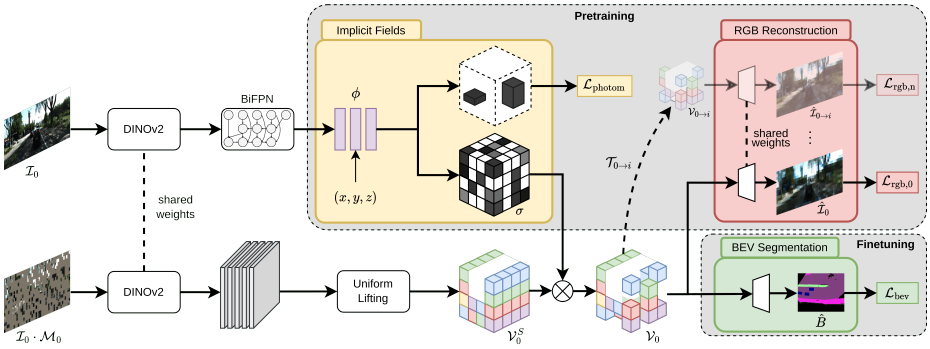


Fig. 2: Overview of LetsMap, our novel unsupervised representation learning framework for label-efficient semantic BEV mapping. The crux of our approach is to leverage FV image sequences to independently model scene geometry and scene representation using two disjoint pathways following an unsupervised training paradigm. The resulting model is finetuned on a small fraction of BEV labels to the task of semantic BEV mapping.

network using the learned high-level understanding of the scene. More recently, foundation models such as DINO [4] and DINOv2 [27] employ self-distillation on large amounts of curated data to learn rich representations of the scene. However, all these approaches work on single timestep images and fail to leverage scene consistency over multiple timesteps. In this work, we explicitly enforce scene consistency over multiple timesteps by proposing a novel temporal masked autoencoding strategy to learn rich scene representations.

3 Technical Approach

In this section, we present an overview of LetsMap, the first unsupervised learning framework for predicting semantic BEV maps from monocular FV images using a label-efficient training paradigm. An overview of our framework is illustrated in Fig. 2. The key idea of our approach is to leverage sequences of multi-camera FV images to learn the two core sub-tasks of semantic BEV mapping, i.e., *scene geometry modeling* and *scene representation learning*, using two disjoint neural pathways following a label-free paradigm, before adapting it to the downstream task in a label-efficient manner. We achieve this desired behavior by splitting the training protocol into sequential FV pretraining and BEV finetuning stages. The FV pretraining stage learns to explicitly model the scene geometry by enforcing scene consistency over multiple views using the photometric loss ($\mathcal{L}_{\text{photom}}$, Sec. 3.2) while learning the scene representation by reconstructing a masked input image over multiple timesteps using the reconstruction loss ($\mathcal{L}_{\text{rgb}}$, Sec. 3.3). Upon culmination of the pretraining phase, the finetuning phase adapts the network to the task of semantic BEV mapping using the cross-entropy loss on the tiny fraction of available BEV labels ($\mathcal{L}_{\text{bev}}$, Sec. 3.4). The total loss of the network is

thus computed as:

$$\mathcal{L} = \begin{cases} \mathcal{L}_{\text{photom}} + \mathcal{L}_{\text{rgb}} & \text{when pretraining} \\ \mathcal{L}_{\text{bev}} & \text{when finetuning} \end{cases}. \tag{1}$$

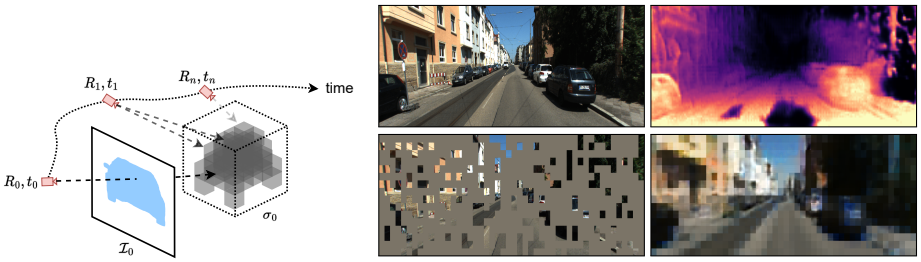
3.1 Network Architecture

Our proposed framework, as shown in Fig. 2, consists of a pretrained DINOv2 [27] (ViT-b) backbone to generate multi-scale features from an input image; a geometry pathway comprising a convolution-based adapter followed by an implicit neural field to predict the scene geometry; a semantic pathway encompassing a sparse convolution-based adapter to capture representation-specific features; an RGB reconstruction head to facilitate reconstruction of the masked input image patches over multiple timesteps; and a BEV semantic head to generate a semantic BEV map from the input monocular FV image during the finetuning phase.

During pretraining, an input image $\mathcal{I}_0$ is processed by the backbone to generate feature maps of three scales. The geometry pathway, $\mathcal{G}$, processes these multi-scale features using a BiFPN [33] layer followed by an implicit field module to generate the volumetric density of the scene at the current timestep. In a parallel branch, a masking module first randomly masks non-overlapping patches in $\mathcal{I}_0$ and the backbone then processes the visible patches to generate the corresponding image features. The semantic pathway $\mathcal{S}$ then generates the representation-specific features using a five-layer adapter that ensures propagation of masked regions using the convolution masking strategy outlined in [34]. We then uniformly lift the resultant 2D features to 3D using the camera projection equation and multiply them with the volumetric density computed from $\mathcal{G}$ to generate scene-consistent voxel features. We warp the voxel grid to multiple timesteps using the ego-motion and collapse it into 2D by applying the camera projection equation along the depth dimension. The RGB reconstruction head then predicts the pixel values for each of the masked patches to reconstruct the image at different timesteps. During finetuning, we disable image masking and orthographically collapse the voxel features along the height dimension to generate the BEV features. A BEV semantic head processes these features to generate semantic BEV predictions.

3.2 Geometric Pathway

The goal of the geometric pathway $\mathcal{G}$ is to explicitly model scene geometry in a label-free manner using only the spatio-temporal images obtained from cameras onboard an autonomous vehicle. Explicit scene geometry modeling allows the network to reason about occlusions and disocclusions in the scene, thus improving the quality of predictions in the downstream task. To this end, we design the task of scene geometry learning using an implicit field formulation wherein the main goal is to estimate the volumetric density of the scene in the camera coordinate system given a monocular FV image, as shown in Fig. 3a. We multiply the estimated volumetric density with the uniformly-lifted semantic features to generate the geometrically consistent semantic features (see Sec. 3.3).



(a) (b)

Fig. 3: (a) An illustration of our neural implicit field module. It leverages spatio-temporal consistency offered by multi-camera images to model scene geometry. (b) FV predictions from our unsupervised pretraining step. A FV image (top left) is processed by the geometry pathway to generate a volumetric density which yields a depth map (top right) upon ray casting. Parallelly, a masked FV image (bottom left) is processed by the semantic pathway to reconstruct the masked image (bottom right).

We generate the volumetric density for the scene by following the idea of image-conditioned NeRF outlined in [38]. Firstly, we retrieve the image features f for randomly sampled points, $\mathbf{x} = (x, y, z)$, along every camera ray by projecting them onto the 2D image plane and computing the value for each projection location using bilinear interpolation. We then pass the image features along with their positional encodings into a two-layer MLP, ϕ, to estimate the volumetric density, $\sigma_{\mathbf{x}}$, at each of the sampled locations. Mathematically, the volumetric density at location $\mathbf{x}$ is computed as:

$$\sigma_{\mathbf{x}} = \phi(f_{\mathbf{u}_{\mathbf{x}}}, \gamma(\mathbf{u}_{\mathbf{x}}, d_{\mathbf{x}})), \tag{2}$$

where $\gamma(\cdot, \cdot)$ denotes the sinusoidal positional encoding computed using the 2D projection $\mathbf{u}_{\mathbf{x}}$ of $\mathbf{x}$ on the image plane and its distance $d_{\mathbf{x}}$ from the camera origin.

During training, we optimize ϕ by first computing the depth map from σ and then computing the photometric loss between the multi-view FV images at both the current as well as future timesteps. Specifically, for a camera ray through pixel location $\mathbf{u}$, we estimate the corresponding depth $\hat{d}_{\mathbf{u}}$ by computing the integral of intermediate depths over the probability of ray termination at a given distance. Accordingly, we sample k points, $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_k$, on each camera ray and compute σ at each of these locations. We then compute the probability of ray termination α_i between every pair of consecutive points $(\mathbf{x}_i, \mathbf{x}_{i+1})$ to determine the distance at which the ray is terminated, i.e., the depth $\hat{d}_{\mathbf{u}}$. Mathematically,

$$\alpha_i = \exp(1 - \sigma_{\mathbf{x}_i} \delta_i), \tag{3}$$

$$\hat{d}_{\mathbf{u}} = \sum_{i=1}^{K} (\prod_{j=1}^{i-1} (1 - \alpha_j)) \alpha_i d_i, \tag{4}$$

where d_i is the distance of $\mathbf{x}_i$ from the camera origin and $\delta_i = d_{i+1} - d_i$. Fig. 3b shows a depth map output from $\mathcal{G}$. We use this depth map to supervise the

geometric pathway $\mathcal{G}$ using the photometric loss between RGB images generated using inverse and forward warping. Inverse warping is described as:

$$I'_{\text{tgt,inv}}(p_{\text{src}}) = I_{\text{tgt}}\langle KT_{\text{src}\to\text{tgt}}d(p_{\text{src}})K^{-1}p_{\text{src}}\rangle, \tag{5}$$

where K is the intrinsic matrix, $\langle\cdot\rangle$ is the bilinear sampling operator, and p_{src} is a pixel coordinate in the source image. Similarly, forward warping is described as:

$$I'_{\text{tgt,fwd}}(KT_{\text{src}\to\text{tgt}}d(p_{\text{src}})K^{-1}p_{\text{src}}) = I_{\text{src}}(p_{\text{src}}), \tag{6}$$

We reduce the impact of occlusions across timesteps from corrupting the photometric loss by only computing the pixelwise minimum for each of the forward and inverse photometric losses. The photometric loss is then computed as:

$$\mathcal{L}_{\text{photom}} = \|I'_{\text{tgt,fwd}} - I_{\text{tgt}}\|_1 + \|I'_{\text{tgt,inv}} - I_{\text{src}}\|_1 \tag{7}$$

3.3 Semantic Pathway

The semantic pathway $\mathcal{S}$ aims to facilitate the learning of holistic feature representations for various scene elements in a label-free manner. This rich pretrained representation enables efficient adaptation to semantic classes during finetuning. To this end, we learn the scene element representations by masking out random patches in the input image and then forcing the network to generate pixel-wise predictions for every masked patch (Fig. 3b). We also exploit the temporal consistency of static elements in the scene by reconstructing the RGB images at future timesteps $t_1, t_2, ..., t_n$ using the masked RGB input at timestep t_0. This novel formulation of temporal masked autoencoding (T-MAE) allows our network to learn spatially- and semantically consistent features which improve its occlusion reasoning ability and accordingly its performance on semantic BEV mapping.

Our semantic pathway $\mathcal{S}$, shown in Fig. 2, masks the input image $\mathcal{I}_0$ using a binary mask M_0 with a masking ratio m, and generates the corresponding masked semantic 3D voxel grid $V_0^{\mathcal{S}}$. We then multiply $V_0^{\mathcal{S}}$ with the volumetric density σ obtained from the geometric pathway $\mathcal{G}$ to generate the intermediate masked voxel grid V_0. During pretraining, we densify V_0 by filling the masked regions using a common mask token `[M]`, and generating pseudo voxel grids $V_{0\to i}$ by warping V_0 using the known camera poses between the current and the i^{th} timesteps. Mathematically,

$$V_{0\to i} = T_{0\to i}V_0, \tag{8}$$

where $T_{0\to i}$ is the transformation between camera poses at timesteps t_0 and t_i. We then independently use the voxel grids $V_0, V_{0\to 1}, V_{0\to 2}, ..., V_{0\to i}$ as inputs to an RGB reconstruction head to reconstruct the RGB images $\hat{\mathcal{I}}_0, \hat{\mathcal{I}}_{0\to 1}, \hat{\mathcal{I}}_{0\to 2}, ..., \hat{\mathcal{I}}_{0\to i}$. We compute the L_2 loss on the normalized pixel values of every patch between $\mathcal{I}_k$ and $\hat{\mathcal{I}}_k$ to generate the supervision for the semantic pathway $\mathcal{S}$. We thus compute the reconstruction loss as:

$$\mathcal{L}_{\text{rgb}} = \sum_{i=0}^{n} \|\mathcal{I}_i^p - \hat{\mathcal{I}}_{0\to i}^p\|_2, \tag{9}$$

where $\mathcal{I}^p$ denotes the per-patch normalized image.

3.4 BEV Finetuning

We set up the network for finetuning by disabling image masking and discarding the RGB reconstruction head. We finetune the network on semantic BEV mapping by training the model on a fraction of BEV ground truth semantic labels using the cross entropy loss function. Mathematically,

$$\mathcal{L}_{\text{bev}} = CE(B, \hat{B}), \tag{10}$$

where B and $\hat{B}$ are the semantic BEV ground truth and semantic BEV prediction masks, respectively.

4 Experimental Results

In this section, we present quantitative and qualitative results of our unsupervised label-efficient semantic BEV mapping framework, LetsMap, and provide extensive ablative experiments to demonstrate the benefit of our proposed contributions.

4.1 Datasets

We evaluate LetsMap on two large-scale autonomous driving datasets, i.e., KITTI-360 [21] and nuScenes [2]. Since neither dataset provides semantic BEV labels, we adopt the label generation pipeline outlined in PoBEV [10] with minor modifications to discard the *occlusion* mask to generate the semantic BEV ground truth labels. We sample one forward-facing perspective image from either fisheye camera for multi-camera supervision in KITTI-360 but use only a single camera in nuScenes due to the lack of sufficient field-of-view overlap between the spatial cameras. For KITTI-360, we hold out sequence 10 for validation and use the remaining 8 sequences for training. For nuScenes, we follow the train-val split from [30] and obtain 702 train and 142 validation sequences.

4.2 Training Protocol

We train LetsMap on images of size 448×1344, and 448×896 for KITTI-360 and nuScenes, respectively. We select these image sizes to ensure compatibility with both the image encoder as well as the lower scales of the BiFPN adapter module since they are divisible by both 14 and 32. The pretraining phase follows a label-free paradigm and trains the network using only spatio-temporal FV images with a window size of 4, masking ratio of 0.75, and masking patch size of 28 for 20 epochs with an initial learning rate (LR) of 0.005 which is decayed by a factor of 0.5 at epoch 15 and 0.2 at epoch 18. We finetune the network on the task of semantic BEV mapping for 100 epochs using only 1% of BEV labels for the KITTI-360 dataset and one sample from every scene for the nuScenes dataset ($\approx \frac{1}{40}\%$). We use an LR of 0.005 during finetuning and decay it by a factor of 0.5 at epoch 75 and 0.2 at epoch 90. We optimize LetsMap using the SGD optimizer with a batch size of 12, momentum of 0.9, and weight decay of 0.0001.

Table 1: Evaluation of semantic BEV mapping on the KITTI-360 dataset. All metrics are reported in [%].

Method	FV	BEV	Road	Side.	Build.	Terrain	Person	2-Wh.	Car	Truck	mIoU
IPM [24]	100%	-	53.03	24.90	15.19	32.31	0.20	0.36	11.59	1.90	17.44
VED [23]	-	100%	65.97	**35.41**	**37.28**	34.34	0.13	0.07	23.83	8.89	25.74
VPN [28]	-	100%	69.90	34.31	33.65	40.17	0.56	2.26	27.76	6.10	26.84
PON [30]	-	100%	67.98	31.13	29.81	34.28	2.28	2.16	37.99	8.10	26.72
PoBEV [10]	-	100%	**70.14**	35.23	34.68	**40.72**	**2.85**	**5.63**	**39.77**	**14.38**	**30.42**
PoBEV [10]	-	1%	60.41	20.97	24.65	23.38	0.15	0.23	21.71	1.23	19.09
SkyEye [9]	100%	1%	69.26	33.48	32.79	39.46	0.00	0.34	32.36	7.93	26.94
LetsMap (Ours)	0%	1%	70.58	34.26	40.68	38.53	1.35	4.74	30.94	10.58	28.96

Table 2: Evaluation of semantic BEV mapping on the nuScenes dataset. All metrics are reported in [%].

Method	FV	BEV	Road	Side.	Manm.	Terrain	Person	2-Wh.	Car	Truck	mIoU
IPM [24]	100%	-	43.51	9.05	26.21	16.60	0.14	0.72	4.65	3.67	13.07
VED [23]	-	100%	67.97	25.23	49.69	31.51	0.80	1.28	21.85	17.51	26.98
VPN [28]	-	100%	66.47	23.94	47.65	33.19	2.02	4.13	22.66	18.33	27.30
PON [30]	-	100%	67.50	24.49	47.02	30.86	2.49	6.85	26.68	18.85	28.09
PoBEV [10]	-	100%	**70.15**	**27.87**	**50.04**	**35.32**	**3.89**	**7.06**	**31.60**	**21.27**	**30.90**
PoBEV [10]	-	$\approx \frac{1}{40}\%$	64.55	19.85	45.21	28.45	1.20	1.06	20.45	11.48	24.03
LetsMap (Ours)	0%	$\approx \frac{1}{40}\%$	**67.72**	**27.06**	**47.10**	**34.78**	**3.31**	**5.79**	**21.92**	**13.57**	**27.66**

4.3 Quantitative Results

We evaluate the performance of LetsMap on KITTI-360 by comparing it with the self-supervised approach SkyEye [9] as well as the fully-supervised baselines outlined in SkyEye. However, since SkyEye cannot be trained on nuScenes due to the lack of FV labels, we compare our approach with only the fully-supervised baselines on the nuScenes dataset. For all experiments, we use the code provided by the authors and ensure fair comparison by using the training protocols described in their original manuscripts. We use the standard mIoU metric for quantifying the performance [15]. Tab. 1 and Tab. 2 present the results of this evaluation for KITTI-360 and nuScenes respectively. For these experiments, we report metrics obtained when fully-supervised approaches are trained using 100% of BEV labels, SkyEye is pretrained using 100% of FV labels and finetuned on a tiny fraction of BEV labels, while LetsMap is trained on only a tiny fraction of BEV labels, i.e., 1% on KITTI-360 and one sample per scene ($\approx \frac{1}{40}\%$) on nuScenes.

We observe from Tab. 1 that our approach, LetsMap, outperforms four of the five fully-supervised baselines by more than 2 pp while using only 1% of BEV labels. Notably, LetsMap also exceeds SkyEye by 2.02 pp without using any additional labeled data. We note that our approach significantly outperforms SkyEye on the static classes of *road* and *building*, as well as the dynamic classes of *person*, *2-wheeler*, and *truck*. This improvement stems from explicit modeling of both scene geometry and representation which ensures well-constrained extents of dynamic objects as well as efficient mapping of scene elements to BEV classes using only 1% of BEV labels. Although better than SkyEye, we observe that

Table 3: Ablation study on the impact of our unsupervised pretraining on the overall network performance. The column "FV" shows whether the models leverage FV pretraining, and the column "PT" denotes whether the models have been pretrained. All experiments are on the KITTI-360 dataset.

BEV	Model	FV	PT	Epochs	Road	Side	Build	Terr.	Pers.	2-Wh.	Car	Truck	mIoU
1%	PoBEV	✗	-	100	60.41	20.97	24.65	23.38	0.15	0.23	21.71	1.23	19.09
	SkyEye	✓	✓		69.26	33.48	32.79	**39.46**	0.00	0.34	**32.36**	7.93	26.94
	LetsMap	✗	✗		69.40	32.09	34.75	35.27	1.01	2.79	28.76	7.66	26.47
	LetsMap	✗	✓		**70.58**	**34.26**	**40.68**	38.53	**1.35**	**4.74**	30.94	**10.58**	**28.96**
5%	PoBEV	✗	-	80	64.45	27.36	30.15	31.66	0.69	0.98	29.75	6.06	23.89
	SkyEye	✓	✓		72.16	37.20	34.89	**42.97**	**4.77**	**9.16**	**40.74**	9.88	31.47
	LetsMap	✗	✗		72.80	37.89	38.59	40.06	2.34	5.62	34.86	16.26	31.05
	LetsMap	✗	✓		**73.74**	**39.56**	**42.07**	41.49	2.46	6.32	34.68	**14.88**	**31.90**
10%	PoBEV	✗	-	50	66.58	30.28	31.76	34.50	1.22	3.28	33.43	7.56	26.08
	SkyEye	✓	✓		73.36	38.30	37.54	**44.62**	**4.80**	**9.67**	**42.84**	10.06	32.65
	LetsMap	✗	✗		74.31	38.45	40.04	41.26	3.19	6.02	35.56	16.53	31.92
	LetsMap	✗	✓		**74.74**	**39.40**	**43.63**	43.33	2.91	6.95	37.62	**18.09**	**33.33**
50%	PoBEV	✗	-	30	69.88	33.81	33.40	40.48	2.47	4.63	38.81	9.84	29.16
	SkyEye	✓	✓		73.10	**39.23**	38.08	**45.72**	**4.05**	**10.44**	**44.72**	12.10	**33.43**
	LetsMap	✗	✗		73.89	38.42	42.25	41.46	2.26	6.26	37.20	15.08	32.10
	LetsMap	✗	✓		**74.29**	38.48	**43.87**	42.77	2.80	5.22	37.68	**15.20**	32.54
100%	PoBEV	✗	-	20	70.14	35.23	34.68	40.72	2.85	5.63	39.77	14.38	30.42
	SkyEye	✓	✓		73.57	**39.45**	38.74	**46.06**	**3.95**	**9.66**	**45.21**	10.92	**33.44**
	LetsMap	✗	✗		74.22	39.39	**42.86**	42.96	2.55	6.66	35.68	**17.11**	32.68
	LetsMap	✗	✓		**74.81**	38.59	42.58	43.67	3.52	6.21	38.47	15.24	32.88

LetsMap underperforms PoBEV for most dynamic classes, reporting 8.83 pp and 3.80 pp lower on *car* and *truck* respectively. This is likely due to insufficient views for training the implicit field or the presence of moving objects which results in its sub-optimal performance. Increasing the number of timesteps and sampling more perspective images from the fisheye cameras could address this limitation.

On the nuScenes dataset, we note that LetsMap is comparable to most of the fully-supervised baselines but is consistently outperformed by the state-of-the-art approach PoBEV. nuScenes, being extremely dynamic and diverse, presents a significant challenge to our implicit field formulation which enforces a static scene constraint. This is especially evident in the *car* and *truck* classes which report 9.68 pp and 7.70 pp lower than PoBEV. Nonetheless, LetsMap is able to efficiently learn the scene representations of static classes, resulting in a comparable performance with all baselines while using only 1% of annotated data.

4.4 Ablation Study

In this section, we investigate the influence of various components of our approach by performing an ablation study on the KITTI-360 dataset. Specifically, we evaluate the impact of model pretraining when presented with varying amounts of labeled BEV data, the benefit of each of our neural pathways, and the effect of varying masking ratios on the overall performance of the network.

Table 4: Ablation study to investigate the efficacy of various network components. All experiments are on the KITTI-360 dataset using 1% of BEV labels.

Model	Geometric	Semantic	Road	Side.	Build.	Terr.	Pers.	2-Wh.	Car	Truck	mIoU
L1	✗	✗	69.40	32.09	34.75	35.27	1.01	2.79	28.76	7.66	26.47
L2	✓	✗	70.85	**34.34**	38.12	35.03	0.93	4.06	29.79	8.84	27.75
L3	✓	✓	70.58	34.26	**40.68**	**38.53**	**1.35**	**4.74**	**30.94**	**10.58**	**28.96**

Impact of Model Pretraining: In this section, we study the impact of model pretraining by finetuning our model *with* and *without* pretraining with varying percentages of labeled BEV data. Accordingly, we establish five percentage splits of BEV labels, i.e., 1%, 5%, 10%, 50%, and 100%, and sample three random sets for each percentage split. We train each percentage split three times, once using each random set, and report the mean value to mitigate the risk of random chance affecting the final results. Moreover, we also train the best two baselines, i.e., PoBEV and SkyEye, across all percentage splits as a reference for evaluating our approach. Tab. 3 presents the results of this ablation study.

We observe that our model trained using our unsupervised pretraining strategy, LetsMap, consistently outperforms our model without pretraining across all percentage splits. The most substantial improvements of 2.49 pp and 1.41 pp occur when finetuning with only 1% and 10% of BEV labels, respectively. We also note that LetsMap outperforms PoBEV by 9.87 pp and SkyEye by 2.02 pp when using only 1% of BEV labels. At extremely low percentage splits, PoBEV does not encounter enough BEV labels to learn the mapping from FV to BEV, while the FV semantic-based pretraining of SkyEye does not impart sufficient geometric modeling and representation learning ability to the network. The notable improvement over SkyEye is primarily attributed to the superior segmentation performance on static classes such as *road* and *building* as well as non-moving dynamic objects such as trucks and buses. This improvement directly stems from the use of implicit neural fields to model scene geometry which helps the network to effectively reason about static elements in the scene. Moreover, we highlight that LetsMap finetuned using only 5% of BEV labels already outperforms the state-of-the-art fully supervised approach PoBEV trained using 100% of BEV labels; thus underscoring the impact of model pretraining in reducing the dependence on large quantities of labeled data. We also note that SkyEye consistently outperforms our approach across four of the five percentage splits on *person*, *two-wheeler*, and *car*. We believe that the superior performance of SkyEye stems from the presence of 100% FV labels which provide unparalleled semantic knowledge during the pretraining phase. Nevertheless, our approach still yields competitive results without using any additional labeled data, thus highlighting the impact of our unsupervised pretraining mechanism.

Influence of Network Components: In this section, we quantify the impact of the geometric and semantic pathways on the overall performance of the network by incrementally incorporating each component into the pretraining step and finetuning the resultant model on 1% of BEV labels. Tab. 4 presents the results

Table 5: Ablation study on the impact of masking ratio. All experiments are on the KITTI-360 dataset using 1% of BEV labels.

Masking Ratio	**0%**	**25%**	**50%**	**75%**	**90%**
mIoU	27.75	27.87	28.22	**28.96**	27.31

of this ablation study. The first row, comprising model L1, illustrates a network without our unsupervised pretraining and serves as a baseline to assess the improvement brought about by the other components. Model L2 incorporates the geometric pathway into the pretraining which results in an improvement of 1.28 pp over model L1. The inclusion of geometric pathway during pretraining allows the implicit field to learn the scene geometry and reason about occlusions which helps improve the IoU metric on most of the classes by nearly 1 pp. Upon the incorporation of our novel temporal MAE strategy via the semantic pathway in model L3, we observe a significant 1.21 pp improvement over model L2. By learning to reconstruct the missing information in the masked patches over multiple timesteps, the network learns spatially- and temporally consistent representations of various scene components which allows it to easily map the learned representation to the semantic BEV task using only 1% of BEV labels.

Impact of Mask Ratios: In this experiment, we evaluate the impact of different masking ratios on the overall performance of the model and present the results in Tab. 5. We observe that a masking ratio of 75% is ideal for our novel temporal masked autoencoding mechanism. Lower masking ratios do not present a sufficiently challenging pretraining task and thus result in only marginal improvements over model L2 in Tab. 4, while higher masking ratios mask out a significant portion of vital information resulting in worse performance as compared to a model with no masked autoencoding.

4.5 Qualitative Results

We qualitatively evaluate the performance of LetsMap in Fig. 4 by comparing it with SkyEye [9] on the KITTI-360 dataset, and PoBEV [10] on the nuScenes dataset. We observe from Fig. 4(a) that both SkyEye and LetsMap are able to predict static classes such as road and sidewalk to a high degree of accuracy, but SkyEye fails to properly localize the car and significantly stretches it along the depth dimension. Our approach, on the other hand, can both properly localize the car in the BEV map as well as predict its extent. In Fig. 4(b) we observe that SkyEye fails to detect the car in the scene, while our approach not only detects the object in the scene but also accurately estimates its extent. Further, we observe in Fig. 4(c) that our approach is able to better predict the extent of the truck as well as predict the location of the *road* class in far-off regions. We hypothesize that our approach efficiently leverages the rich geometric and semantic knowledge learned by the disjoint neural pathways to effectively transfer the knowledge from FV to BEV even when using only 1% of BEV labels. We observe from Fig. 4(d-f)

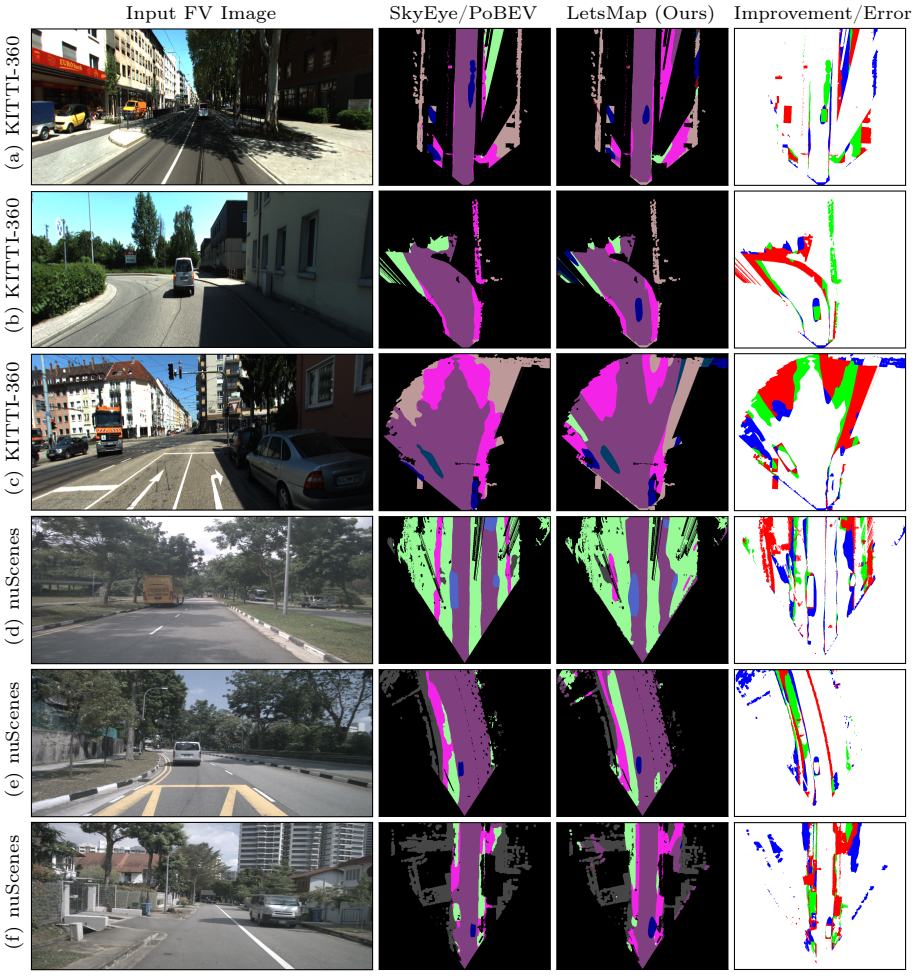


Fig. 4: Qualitative results of our unsupervised learning framework LetsMap in comparison with SkyEye [9] on the KITTI-360 dataset, and PoBEV [10] on the nuScenes dataset. We also display an Improvement/Error map which highlights pixels where LetsMap outperforms the baseline in green, where baseline outperforms LetsMap in blue, and where both models misclassify in red.

that our model accurately estimates the static elements of the scene which is evident from the error/improvement map in the last column. Although trained on only 1% of BEV labels as compared to 100% for PoBEV, our approach manages to precisely capture the locations of *car* and *truck* instances in the BEV map. Fig. 4(d) also highlights one of the limitations of our approach wherein dynamic objects such as cars, trucks, and pedestrians are often radially stretched. This limitation is primarily caused by the lack of sufficient camera views to learn the entire 3D representation of the dynamic objects which could be addressed by

exploiting the cross-view correlation and spatio-temporal consistency of surround view cameras. Interestingly, Fig. 4(e) reveals that LetsMap is also able to leverage representative scene priors learned during the pretraining step to infer knowledge about occluded regions. In this prediction, LetsMap is able to predict that the *terrain* class extends further into the occluded region, unlike PoBEV which incorrectly predicts this region as *sidewalk*; thus highlighting the benefit of our unsupervised pretraining protocol.

4.6 Discussion of Limitations

LetsMap suffers from three main limitations, all of which stem from modeling scene geometry using implicit fields. Firstly, existing implicit field formulations enforce a strong static scene assumption which is often violated in real-world autonomous driving environments. An explicit dynamic object handling module as discussed in [16] could be used to address this limitation in such environments. Secondly, implicit fields rely on a large and diverse set of camera views of a given object to learn its optimal scene geometry. However, this is infeasible in autonomous driving which results in the generation of sub-optimal volumetric grids in their current form. Lastly, our formulation of implicit fields is supervised using the photometric loss between temporal multi-camera images. However, the photometric loss is often sensitive to varying lighting conditions, occlusions, and disocclusions, as well as object motion - all of which are exacerbated when the ego motion between two frames is large. This problem can typically be addressed by adding a stereo camera to capture slightly offset images and provide a reliable frame for loss computation.

5 Conclusion

In this paper, we present the first unsupervised representation learning approach, LetsMap, for predicting semantic BEV maps from monocular FV images using a label-efficient learning paradigm. Our approach leverages the spatio-temporal consistency and rich scene semantics offered by FV image sequences to independently learn the sub-tasks of BEV mapping, i.e., scene geometry estimation and scene representation learning, in an unsupervised pretraining step. It finetunes the resultant model on the BEV segmentation task using only a small fraction of labels in BEV. Using extensive evaluations on the KITTI-360 and nuScenes datasets, we demonstrate that LetsMap performs on par with the existing fully-supervised and self-supervised approaches while using only 1% of BEV labels and without relying on any additional source of labeled supervision.

Acknowledgements

This work was partially funded by Qualcomm Technologies Inc. and a hardware grant from NVIDIA.

References

1. Bustos, A.P., Chin, T.J., Eriksson, A., Reid, I.: Visual SLAM: Why Bundle Adjust? In: Int. Conf. on Robotics & Automation. pp. 2385–2391 (2019)
2. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: IEEE Conf. on Computer Vision and Pattern Recognition. pp. 11621–11631 (2020)
3. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: Proc. of the Conf. on Neural Information Processing Systems (NIPS) (2020)
4. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Int. Conf. on Computer Vision. pp. 9650–9660 (2021)
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: Int. Conf. on Machine Learning. vol. 119, pp. 1597–1607 (2020)
6. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. IEEE Trans. on Pattern Analysis and Machine Intelligence **32**(8), 1362–1376 (2009)
7. Gao, S., Wang, Q., Sun, Y.: S2g2: Semi-supervised semantic bird-eye-view grid-map generation using a monocular camera for autonomous driving. IEEE Robotics & Automation Letters (2022)
8. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: Int. Conf. on Learning Representations (2018)
9. Gosala, N., Petek, K., Drews-Jr, P.L.J., Burgard, W., Valada, A.: Skyeye: Self-supervised bird's-eye-view semantic mapping using monocular frontal view images. In: IEEE Conf. on Computer Vision and Pattern Recognition. pp. 14901–14910 (June 2023)
10. Gosala, N., Valada, A.: Bird's-eye-view panoptic segmentation using monocular frontal view images. IEEE Robotics & Automation Letters **7**(2), 1968–1975 (2022)
11. Harley, A.W., Fang, Z., Li, J., Ambrus, R., Fragkiadaki, K.: Simple-bev: What really matters for multi-sensor bev perception? In: Int. Conf. on Robotics & Automation. pp. 2759–2765 (2023)
12. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: IEEE Conf. on Computer Vision and Pattern Recognition. pp. 16000–16009 (2022)
13. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: IEEE Conf. on Computer Vision and Pattern Recognition (2020)
14. Hindel, J., Gosala, N., Bregler, K., Valada, A.: Inod: Injected noise discriminator for self-supervised representation learning in agricultural fields. IEEE Robotics & Automation Letters (2023)
15. Hurtado, J.V., Valada, A.: Semantic scene segmentation for robotics. In: Deep Learning for Robot Perception and Cognition, pp. 279–311. Elsevier (2022)
16. Ingale, A.K., et al.: Real-time 3d reconstruction techniques applied in dynamic scenes: A systematic literature review. Computer Science Review **39**, 100338 (2021)
17. Irshad, M.Z., Zakharov, S., Liu, K., Guizilini, V., Kollar, T., Gaidon, A., Kira, Z., Ambrus, R.: Neo 360: Neural fields for sparse view synthesis of outdoor scenes. In: Int. Conf. on Computer Vision (2023)
18. Kulkarni, N., Johnson, J., Fouhey, D.F.: Directed ray distance functions for 3d scene reconstruction. In: European Conf. on Computer Vision. pp. 201–219 (2022)

19. Lang, C., Braun, A., Schillingmann, L., Haug, K., Valada, A.: Self-supervised representation learning from temporal ordering of automated driving sequences. IEEE Robotics & Automation Letters (2024)
20. Li, Q., Wang, Y., Wang, Y., Zhao, H.: HDMapNet: An online HD map construction and evaluation framework. In: Int. Conf. on Robotics & Automation. pp. 4628–4634 (2022)
21. Liao, Y., Xie, J., Geiger, A.: Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. IEEE Trans. on Pattern Analysis and Machine Intelligence (2022)
22. Liu, Z., Tang, H., Amini, A., Yang, X., Mao, H., Rus, D.L., Han, S.: Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation pp. 2774–2781 (2023)
23. Lu, C., van de Molengraft, M.J.G., Dubbelman, G.: Monocular semantic occupancy grid mapping with convolutional variational encoder–decoder networks. IEEE Robotics & Automation Letters **4**(2), 445–452 (2019)
24. Mallot, H.A., Bülthoff, H.H., Little, J., Bohrer, S.: Inverse perspective mapping simplifies optical flow computation and obstacle detection. Biological cybernetics **64**(3), 177–185 (1991)
25. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) European Conf. on Computer Vision. pp. 405–421 (2020)
26. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: European Conf. on Computer Vision (2016)
27. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
28. Pan, B., Sun, J., Leung, H.Y.T., Andonian, A., Zhou, B.: Cross-view semantic segmentation for sensing surroundings. IEEE Robotics and Automation Letters **5**(3), 4867–4873 (2020)
29. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: European Conf. on Computer Vision (2020)
30. Roddick, T., Cipolla, R.: Predicting semantic map representations from images using pyramid occupancy networks. In: IEEE Conf. on Computer Vision and Pattern Recognition (2020)
31. Saha, A., Mendez, O., Russell, C., Bowden, R.: Translating images into maps. In: Int. Conf. on Robotics & Automation. pp. 9200–9206 (2022)
32. Schramm, J., Vödisch, N., Petek, K., Kiran, B.R., Yogamani, S., Burgard, W., Valada, A.: Bevcar: Camera-radar fusion for bev map and object segmentation. arXiv preprint arXiv:2403.11761 (2024)
33. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: IEEE Conf. on Computer Vision and Pattern Recognition. pp. 10781–10790 (2020)
34. Tian, K., Jiang, Y., Diao, Q., Lin, C., Wang, L., Yuan, Z.: Designing BERT for convolutional networks: Sparse and hierarchical masked modeling. In: The Eleventh International Conference on Learning Representations (2023)
35. Vödisch, N., Cattaneo, D., Burgard, W., Valada, A.: Continual slam: Beyond lifelong simultaneous localization and mapping through continual learning. arXiv preprint arXiv:2203.01578 (2022)
36. Wimbauer, F., Yang, N., Rupprecht, C., Cremers, D.: Behind the scenes: Density fields for single view reconstruction pp. 9076–9086 (2023)

37. Yang, C., Chen, Y., Tian, H., Tao, C., Zhu, X., Zhang, Z., Huang, G., Li, H., Qiao, Y., Lu, L., Zhou, J., Dai, J.: Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In: IEEE Conf. on Computer Vision and Pattern Recognition. pp. 17830–17839 (2023)
38. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: IEEE Conf. on Computer Vision and Pattern Recognition. pp. 4578–4587 (2021)