Drive&Segment: Unsupervised Semantic Segmentation of Urban Scenes via Cross-modal Distillation

Antonin Vobecky[1,2], David Hurych[2], Oriane Siméoni[2], Spyros Gidaris[2], Andrei Bursuc[2], Patrick Pérez[2], and Josef Sivic[1]

[1] Czech Institute of Informatics, Robotics and Cybernetics, CTU in Prague
[2] valeo.ai

Abstract. This work investigates learning pixel-wise semantic image segmentation in urban scenes without any manual annotation, just from the raw non-curated data collected by cars which, equipped with cameras and LiDAR sensors, drive around a city. Our contributions are threefold. First, we propose a novel method for cross-modal unsupervised learning of semantic image segmentation by leveraging synchronized LiDAR and image data. The key ingredient of our method is the use of an object proposal module that analyzes the LiDAR point cloud to obtain proposals for spatially consistent objects. Second, we show that these 3D object proposals can be aligned with the input images and reliably clustered into semantically meaningful pseudo-classes. Finally, we develop a cross-modal distillation approach that leverages image data partially annotated with the resulting pseudo-classes to train a transformer-based model for image semantic segmentation. We show the generalization capabilities of our method by testing on four different testing datasets (Cityscapes, Dark Zurich, Nighttime Driving and ACDC) without any finetuning, and demonstrate significant improvements compared to the current state of the art on this problem. [3]

Keywords: autonomous driving · unsupervised semantic segmentation

1 Introduction

In this work, we investigate whether it is possible to learn pixel-wise semantic image segmentation of urban scenes without the need for any manual annotation, just from the raw non-curated data that are collected by cars equipped with cameras and LiDAR sensors while driving in town. This topic is important as current methods require large amounts of pixel-wise annotations over various driving conditions and situations. Such a manual segmentation of images on a large scale is very expensive, time consuming, and prone to biases.

Currently, the best methods for unsupervised learning of semantic segmentation assume that images contain centered objects [48] rather than full scenes, or use spatial self-supervision available in the image domain [15]. They do not leverage additional modalities, such as the LiDAR data, available for urban scenes in the autonomous driving set-ups. In this work, we develop an approach for unsupervised semantic segmentation that learns to segment complex scenes containing many objects, including thin

[3] See project webpage https://vobecant.github.io/DriveAndSegment/ for the code and more.

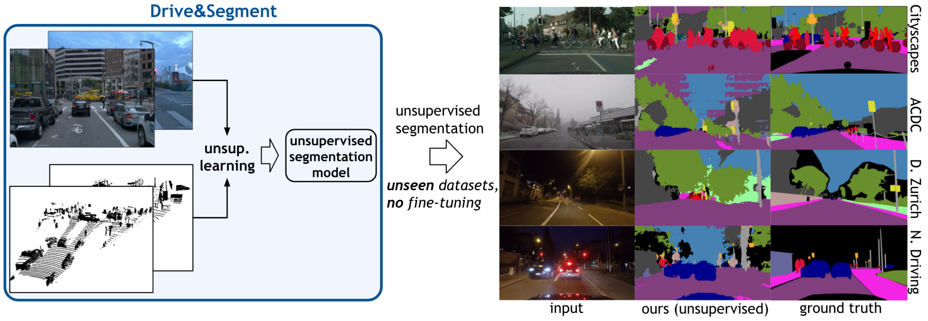


Fig. 1. Proposed fully-unsupervised approach. From uncurated images and LiDAR data, our Drive&Segment approach learns a semantic image segmentation model with no manual annotations. The resulting model performs unsupervised semantic segmentation of new unseen datasets without any human labeling. It can segment complex scenes with many objects, including thin structures such as people, bicycles, poles or traffic lights. Black denotes the ignored label.

structures such as pedestrians or traffic lights, without the need for any manual annotation, but instead leveraging cross-modal information available in (aligned) LiDAR point clouds and images, see Fig. 1. Exploiting point clouds as a form of supervision is, however, not straightforward: data from LiDAR and camera are rarely perfectly synchronized; moreover, point clouds are unstructured and of much lower resolution compared to RGB images; finally, extracting semantic information from LiDAR is still a very hard problem. In this work, we overcome these issues and demonstrate that it is nevertheless possible to extract useful pixel-wise semantic supervision from LiDAR data.

The contributions of our work are threefold. First, we propose a novel method for cross-modal unsupervised learning of semantic image segmentation by leveraging synchronized LiDAR and image data. The key ingredient is a module analyzing the LiDAR point cloud to obtain proposals for spatially consistent objects that can be clearly separated from each other and from the ground plane in the 3D scene. Second, we show that these 3D object proposals can be aligned with input images and reliably clustered into semantically meaningful pseudo-classes by using image features from a network trained without supervision. We demonstrate that this approach is robust to noise in point clouds and delivers, without the need for any manual annotation, pseudo-classes with pixel-wise segmentation for a variety of objects present in driving scenes. These classes include objects such as pedestrians or traffic lights that are notoriously hard to segment automatically in the image domain. Third, we develop a novel cross-modal distillation approach that first trains a teacher network with the available partial pseudo labels, and then exploits its predictions for training the student with pixel-wise pseudo annotations that cover the whole image. In addition, our approach exploits geometric constraints extracted from the LiDAR point cloud during the teacher-student learning process to refine the teacher predictions that are distilled into the student network. Implemented with transformer-based networks, this cross-modal distillation approach results in a trained student model that performs well in a variety of challenging conditions such as day, night, fog, or rain, outside the domain of the original training dataset, as shown in Fig. 1.

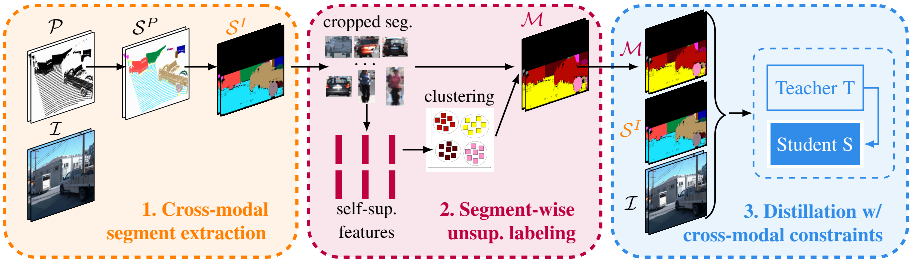


Fig. 2. Overview of Drive&Segment. We first perform cross-modal segment extraction on training dataset by exploiting raw *LiDAR* point clouds $\mathcal{P}$ and raw *images* $\mathcal{I}$. This yields segments $\mathcal{S}^I$ projected onto the image space (§ 3.1). By clustering their self-supervised features, we obtain an unsupervised labeling of these segments (§ 3.2) and, as a consequence, of their pixels. This provides pixel-wise *pseudo ground truth* for the next learning step. Finally, given the pseudo-labels and the segments, we perform distillation with cross-modal constraints (§ 3.3) that conjugates information of the LiDAR and the images to learn a final segmentation model using a teacher-student architecture. The learnt segmentation model S –highlighted in the figure– is used for inference on unseen datasets, yielding compelling results (§ 4).

We train our proposed unsupervised semantic segmentation method on Waymo Open [45] and nuScenes [8] datasets (nuScenes results are in the appendix), and test it on four different datasets in the autonomous driving domain, Cityscapes [16], Dark-Zurich [42], Nighttime driving [17] and ACDC [43]. We demonstrate significant improvements compared to the current state of the art, improving the current best published unsupervised semantic segmentation results on Cityscapes from 15.8 to 21.8 and from 4.6 to 14.2 on Dark Zurich, measured by mean intersection over union.

2 Related work

In this work, we investigate the task of *semantic image segmentation with no human supervision*. We discuss the corresponding prior art below.

Image semantic segmentation. Semantic segmentation is a challenging key visual perception task, especially for autonomous driving [16,37,43,49,54]. Current top-performing models are based on fully convolutional networks [34] with encoder-decoder structures and a large diversity of designs [12,14,33,41,50,59]. Recent progress in vision transformers (ViT) [19] opened the door for a new wave of decoders [44,52,55,60] with appealing performance. All methods, in particular transformer-based [19], attain impressive performance by exploiting large amounts of pixel-wise labeled data. Yet, urban scenes are expensive to annotate manually (1.5h-3h per image [16,43]). This motivates recent works to rely less on pixel-wise supervision.

Reducing supervision for semantic segmentation. A popular strategy when dealing with limited labeled data is to pre-train some of the blocks of the architecture, e.g., the encoder, on related auxiliary tasks with plentiful labels [18,56]. Pre-training encoder for image classification, e.g., on ImageNet [18], has been shown to be a successful recipe for both convnets [12] and ViT-based models [44]. Pre-training can be conducted even without any human annotations on artificially-designed self-supervised pretext tasks [9,22,23,24,25,27] with impressive results on a variety of downstream tasks. Fully unsupervised semantic segmentation has been recently addressed [6,13,15,28,30,31,38,48,57] via generative models for generating object masks [6,13,38] or self-supervised clustering [15,30]. Prior methods are limited to segmenting foreground objects of a single class [6,13] or to *stuff* pixels that outnumber by far *things* pixels [30,38]. Others assume that images contain centered objects [48], rely on weak spatial cues from the image domain [13,15,30] or require instance masks during pre-training and annotated data at test time [28]. In contrast, our approach exploits cross-modal supervision from aligned LiDAR point clouds and images. We show that leveraging this information can considerably improve segmentation performance in complex autonomous driving scenes with multiple classes and strong class imbalance, outperforming PiCIE [15], the current state of the art in unsupervised segmentation.

Cross-modal self-supervised learning. Leveraging language, vision, and/or audio, self-supervised representation learning has seen tremendous progress in recent years [2,3,4,36,39,40,58]. Besides learning useful representations, these approaches show that signals from one modality can help train object detectors in the other, e.g., detecting instruments that sound in a scene [11,39,58], and even other object types [1]. In autonomous driving, a vehicle is equipped with diverse sensors (e.g., camera, LiDAR, radar) and cross-modal self-supervision is often used to generate labels from a sensor for augmenting the perception of another [5,29,46,51]. LiDAR clues [46] have been recently shown to improve unsupervised object detection. In contrast, we consider the problem of pixel-wise unsupervised semantic segmentation, a particularly challenging task given the sparsity and low resolution of LiDAR point clouds.

3 Proposed unsupervised semantic segmentation

Our goal is to train an *image segmentation model* with *no human annotation*, by exploiting easily-available aligned *LiDAR* and *image* data. To that end, we propose a novel method, Drive&Segment, that consists of three major steps and is illustrated in Figure 2. First, as discussed in Section 3.1, we extract *segment* proposals for the objects of interest from 3D LiDAR point clouds and project them to the aligned RGB images. In the second step, presented in Section 3.2, we build *pseudo-labels* by clustering *self-supervised* image features corresponding to these segments. Finally, in Section 3.3, we propose a new teacher-student training scheme that incorporates *spatial constraints* from the LiDAR data and learns an unsupervised segmentation model from the noisy and partial pseudo-annotations generated in the previous two steps.

3.1 Cross-modal segment extraction

Throughout the next sections, we consider a dataset composed of a set $\mathcal{P}$ of 3D point clouds and a set $\mathcal{I}$ of images aligned with the point clouds. In this section, we detail the process for extracting segments of interest in an image $\mathbf{I} \in \mathcal{I}$ using the corresponding aligned LiDAR point cloud $\mathbf{P} \in \mathcal{P}$. The process, illustrated in Fig. 3, consists of three major steps. We start by segmenting the LiDAR point cloud $\mathbf{P}$ using its geometrical

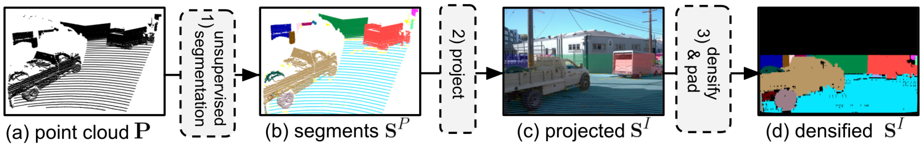

(a) point cloud $\mathbf{P}$
1) unsupervised segmentation
(b) segments $\mathbf{S}^P$
2) project
(c) projected $\mathbf{S}^I$
3) densify & pad
(d) densified $\mathbf{S}^I$

Fig. 3. Cross-modal segment extraction. Input raw point cloud (a) is first segmented with [7] into object segment candidates (b), which are then projected into the image (c); Projected segments are densified to get pixel-level pseudo-labels, with missed pixels being labeled as "ignore", as shown in black (d).

properties. Then, we project the resulting 3D segments into the image $\mathbf{I}$, and densify the output to obtain pixel-level segments.

Geometric point cloud segmentation. We first extract J non-overlapping object segmentation proposals (*segments*), from the LiDAR point cloud $\mathbf{P}$. Let $\mathbf{S}^P = \{s_j^P\}_{j=1}^J$ be this set, where each segment s_j^P is a subset of the 3D point cloud $\mathbf{P}$ and, $\forall j \neq j'$, $s_j^P \cap s_{j'}^P = \varnothing$. Additionally, we refer to the set of segments over the entire dataset as $\mathcal{S}^P$, with $\mathbf{S}^P \subset \mathcal{S}^P$. The J segments detected in one point cloud should ideally correspond to J individual objects in the scene. To get them, we use the unsupervised 3D point cloud segmentation proposed in [7], which exploits the geometrical properties of point clouds and range images. [4] It is a two-stage process that segments the ground plane and objects using greedy labeling by breadth-first search in the range image domain. Urban scenes are particularly suited to this purely geometry-based method as most objects are spatially well separated and the ground plane is relatively easy to segment out.

Point-cloud-to-image transfer. The next step of the segment extraction is to transfer the set $\mathbf{S}^P$ of point cloud segments to the image $\mathbf{I}$, producing the set $\mathbf{S}^I$. Despite LiDAR data and camera images being captured at the same time, one-to-one matching is not straightforward. Indeed, among other difficulties, LiDAR data only covers a fraction of the image plane because of its different field of view, its lower density and its lack of usable measurements on far away objects or on the sky for instance. To overcome the mismatch between the two modalities, we proceed as follows. First, we project the points from the point cloud to the image using the known sensors' calibration. This gives us the locations of 3D points from the point cloud in the image. We also identify locations with invalid measurements in the LiDAR's range image, e.g., reflective surfaces or the sky, and assign an "ignore" label to the respective locations.

Densify & pad. Next, we perform nearest-neighbor interpolation to propagate the $J+1$ segment labels to all pixels, where J is the number of segments (ideally corresponding to objects) and $+1$ denotes the additional "ignore" label. Last, we pad the image with "ignore" label to the input image size.

3.2 Segment-wise unsupervised labeling

Next, the objective is to produce *pseudo-labels* for all extracted segments of interest in the image space, and to do so without any supervision. In particular, we leverage the

[4] Range images are depth maps corresponding to the raw LiDAR measurements. Valid measurements are back-projected to the 3D space to form a point cloud.

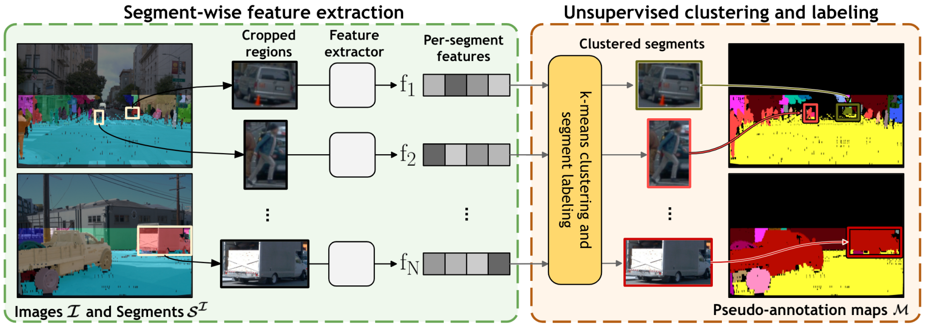

Segment-wise feature extraction
Cropped regions
Feature extractor
Per-segment features
f_1
f_2
f_N
Images $\mathcal{I}$ and Segments $\mathcal{S}^{\mathcal{I}}$
Unsupervised clustering and labeling
Clustered segments
k-means clustering and segment labeling
Pseudo-annotation maps $\mathcal{M}$

Fig. 4. Segment-wise unsupervised pseudo-labeling. First, given object segments $\mathcal{S}^{\mathcal{I}}$ obtained in the segment extraction stage (left), we take crops around all N objects and feed them to a feature extractor to get a set of N feature vectors. Then, we use the k-means algorithm to cluster the feature vectors into k clusters. Finally, we assign pixel-wise *pseudo-labels* to all pixels belonging to each segment based on the corresponding cluster id. Pixels not covered by a segment are assigned the label "ignore" (black).

very recent ViT [19] model pre-trained in a fully unsupervised fashion [10] which has shown impressive results on a range of downstream tasks. We use this representation for unsupervised learning of pseudo-labels as described next and illustrated in Figure 4.

Considering here the image $\mathbf{I}$, we crop a tight rectangular region in the image around each segment $s_j^I \in \mathbf{S}^I$ obtained using the proposal mechanism described in the previous section. We resize it and feed it to the ViT model to extract the feature $\mathbf{f}_j$ corresponding to the output features of the `CLS` token. To limit the influence of pixels outside the object segment, which may correspond to other objects or the background, we mask out these pixels before computing the features. We repeat this operation for all segments in each image $\mathbf{I}$ in the training dataset and cluster the `CLS` token features using k-means algorithm, thus discovering k clusters of visually similar segments. Therefore, each feature $\mathbf{f}_j$ and its corresponding segment s_j^I, is assigned a cluster id l_j in $[\![1, k]\!]$.

To get a dense *segmentation* map $\mathbf{M}$ corresponding to the image $\mathbf{I}$, we assign discovered cluster ids to each pixel belonging to a segment in the image. We additionally assign a predefined *ignore* label to pixels not covered by segments, which correspond to missing annotations. This allows us to construct a set $\mathcal{M}$ of dense *maps of pseudo-annotations*, that we later use as a pseudo-ground-truth. Examples of resulting segmentation maps are shown in Figure 4.

3.3 Distillation with cross-modal spatial constraints

After previous steps, we now have a set of pseudo-annotated segmentation maps $\mathbf{M} \in \mathcal{M}$, one for every image $\mathbf{I}$ in the training dataset. However, as explained above, the pseudo-annotations are **only partial**. This is because the segments that were used to construct them do not cover all pixels of an image. Furthermore, due to imperfections in the segment extraction process or the segment clustering step, these annotations are noisy. Therefore, directly training an image segmentation model with those pseudo-labels might be sub-optimal. Instead, we propose a new teacher-student training ap-

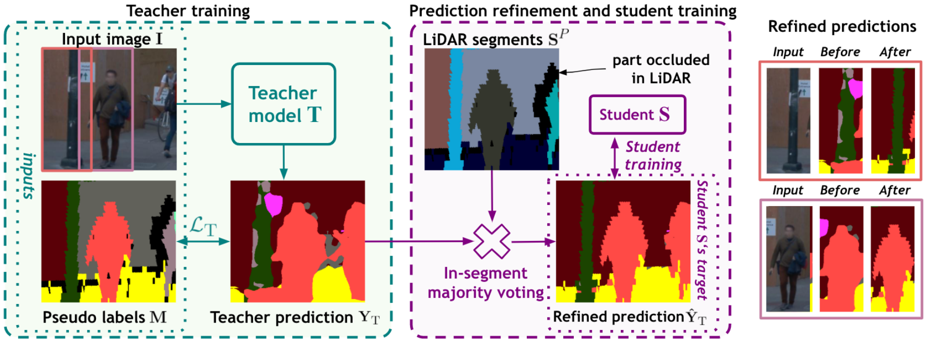


Fig. 5. Teacher prediction refinement using spatial constraints. First, the teacher T is trained using loss $\mathcal{L}_{\mathrm{T}}$ on images in $\mathcal{I}$ together with segmentation maps in $\mathcal{M}$ obtained from segment-wise unsupervised pseudo-labeling. The teacher predictions $\mathbf{Y}_{\mathrm{T}}$ are refined, using LiDAR segments $\mathbf{S}^P$, into maps $\hat{\mathbf{Y}}_{\mathrm{T}}$ that are then used to train the student. Note that teacher's predictions span the whole image, producing outputs even in areas where LiDAR segments $\mathbf{S}^P$ are not available.

proach with cross-modal distillation, which is able to learn more accurate unsupervised segmentation models under such partial and noisy pseudo-annotations.

Training the teacher. The first step of our teacher-student approach is to train the teacher T to make pixel-wise predictions only on the pixels for which pseudo-annotations are available, i.e., only for the pixels that belong to a segment. We denote $\mathbf{Y}_{\mathrm{T}} = \mathrm{T}(\mathbf{I}) \in \mathbb{R}^{H \times W}$ the segmentation predictions made by the teacher model on image $\mathbf{I}$ with a resolution of $H \times W$ pixels. We train the teacher T using loss $\mathcal{L}_{\mathrm{T}}(\mathbf{I})$ on image $\mathbf{I}$:

$$\mathcal{L}_{\mathrm{T}}(\mathbf{I}) = \frac{1}{\sum_{h,w} B_{(h,w)}} \sum_{h,w} \mathrm{CE}\left(\mathbf{Y}_{\mathrm{T},(h,w)}, \mathbf{M}_{(h,w)}\right) B_{(h,w)}, \tag{1}$$

where CE is the cross-entropy loss measuring the discrepancy between the predicted labels $\mathbf{Y}_{\mathrm{T}}$ and target pseudo-labels $\mathbf{M}$ for each pixel (h, w), and B is a $H \times W$ binary mask for filtering out pixels without pseudo-annotations. The loss is normalized w.r.t. the number of pseudo-labeled pixels in the image. The trained teacher T is then able to predict pixel-wise segmentation for all pixels in an image, even if they do not belong to a segment. Moreover, since the teacher T is trained on a large set of pseudo-annotated segments, it learns to smooth out some of the noise in the raw pseudo-annotations.

Integrating spatial constraints. Considering this smoothing property, we can exploit the trained teacher T for generating new, complete (instead of partial) and smooth pseudo-segmentation maps for the training images. In addition, we propose to refine these teacher-generated pseudo-segmentation maps by using the projected LiDAR segments; indeed, these segments encode useful 3D spatial constraints as they often correspond to complete 3D objects, thus respecting the depth discontinuities and occlusion boundaries. In particular, for each image segment s_j^I in image $\mathbf{I}$, we apply majority voting to pixel-wise teacher predictions $\mathbf{Y}_{\mathrm{T}}$ inside the segment. Then, we annotate each pixel belonging to the segment with its most frequently predicted label, giving us a new refined segmentation map $\hat{\mathbf{Y}}_{\mathrm{T}} \in \mathbb{R}^{H \times W}$. This procedure is illustrated in Figure 5.

Training the student. Having computed these complete, teacher-generated and spatially-refined pseudo-segmentation maps $\hat{\mathbf{Y}}_{\mathrm{T}}$, we train a student network S using the following loss

$$\mathcal{L}_{\text{distill}}(\mathbf{I}) = \frac{1}{HW} \sum_{h,w} \mathrm{CE}\left(\hat{\mathbf{Y}}_{\mathrm{T},(h,w)}, \mathbf{Y}_{\mathrm{S},(h,w)}\right), \tag{2}$$

where the cross-entropy is computed between $\hat{\mathbf{Y}}_{\mathrm{T}}$ and the segmentation map $\mathbf{Y}_{\mathrm{S}} \in \mathbb{R}^{H \times W}$ predicted by the student at the same resolution as the teacher. The outputs of the trained student are our final unsupervised image segmentation predictions. Further details about our training can be found in Section 4.1.

4 Experiments

In this section, we give the implementation details, compare our results with the state-of-the-art unsupervised semantic segmentation methods on four different datasets, and ablate the key components of our approach.

4.1 Experimental setting

Methods and architectures. We investigate the benefits of our approach using two different semantic segmentation models to demonstrate the generality of our method. We implement Drive&Segment with both a classical convolutional model and a transformer-based architecture. For the convolutional architecture, we follow [15] and use a ResNet18 [26] backbone followed by an FPN [33] decoder. For the transformer-based architecture, we use the state-of-the-art Segmenter [44] model. We use the ViT-S/16 [19] model as the Segmenter's encoder and use a single layer of the mask transformer [44] as a decoder. We compare our method to three recent unsupervised segmentation models: IIC [30], modified version of DeepCluster [9] (DC), and PiCIE [15]. Please refer to [15] for implementation details of IIC and DC.

Training. In the following, we first discuss how we obtain segment labels by k-means clustering, then we talk about details of pre-training the model backbones, which is followed by the discussion of the datasets for actual training of the models. Finally, we give details of the training procedure.

K-means. We use $k = 30$ in the k-means algorithm (the ablation of the value of k is in the appendix Section A.2). To extract segment-wise features used for k-means clustering, we use `CLS` token features of the DINO-trained [10] ViT-S [19] model. The ablation of the feature extractor is in Section 4.3. Obtained segment-wise labels serve as pseudo-annotations for training the ResNet18-FPN and Segmenter models, as discussed in Section 3.2.

Pre-training data and networks. To be directly comparable to [15], in our experiments with the ResNet18+FPN model, we initialize its backbone with a ResNet18 trained with supervision on the ImageNet-1k [18] classification task, exactly as all the compared prior methods (PiCIE, DC, and IIC). However, we aim at having a completely unsupervised setup. Therefore, in our experiments with Segmenter, we initialize

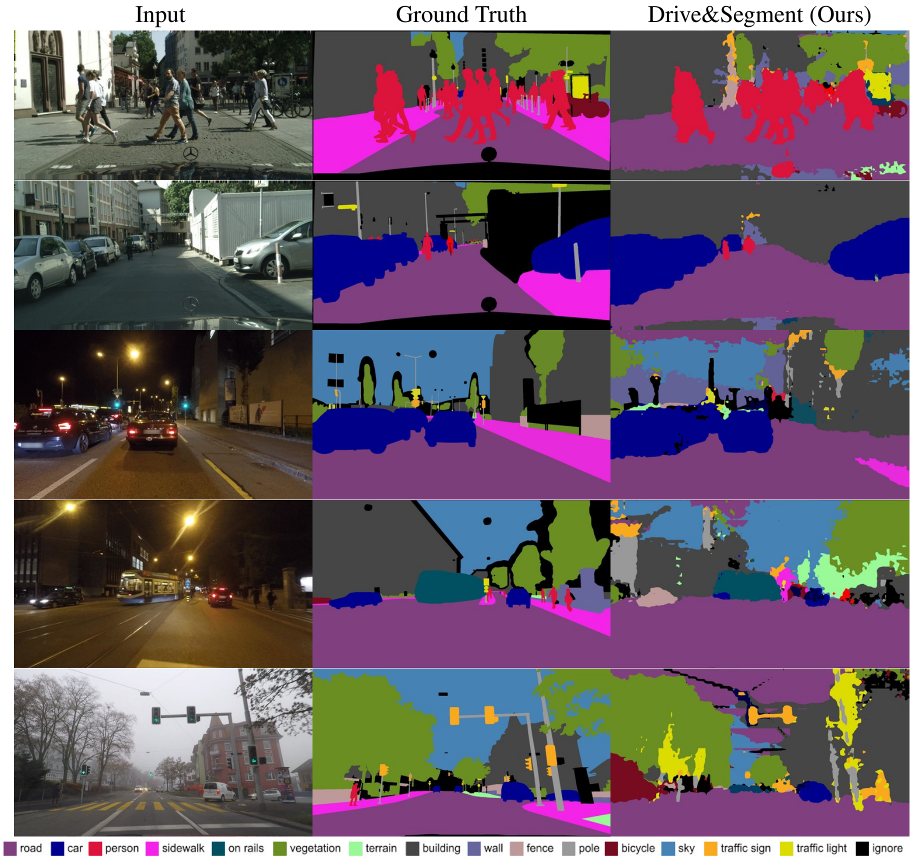


Fig. 6. Qualitative results for *unsupervised* semantic segmentation using our Drive&Segment approach. To obtain the best matching between our pseudo-classes and the set of ground-truth classes, we use the Hungarian algorithm. The first two rows show samples from the Cityscapes [16] dataset, and the other three rows show samples from the night and fog splits of the ACDC [43] dataset. Please see Figures 12, 13 and 14 in the appendix for more qualitative results.

its backbone with a ViT-S also trained on ImageNet-1k [18] but with the self-supervised DINO approach [10]. Please note that for both backbones we use the same pre-training data (ImageNet-1k) exactly as the prior methods PiCIE, DC, and IIC with which we compare. These pre-trained backbones are then used as an initialization for training the actual segmentation models on specific autonomous driving datasets as described next.

Training datasets. We train our models on about 7k images from the Waymo Open [45] dataset, which has both image and LiDAR data available. For the baseline methods (IIC [30], modified DC [9] and PiCIE [15]) we follow the setup from [15], i.e., we train the models on all available images of Cityscapes [16], meaning the 24.5k images from the *train*, *test*, and *train_extra* splits. Note that those models then do not face the problem of domain gap when evaluated on the Cityscapes [16] dataset. To be directly comparable with our approach, we also train a variant of modified DC [9] and

PiCIE [15] on the same subset of the Waymo Open [45] dataset as used in our approach. Furthermore, to test the generalizability of our method to other training datasets, we provide results of modified DC, PiCIE, and our Drive&Segment approach when trained on the nuScenes [8] dataset in Sec. A.1 in the appendix.

Optimization. To train IIC [30], modified DC [9], and PiCIE [15], we use the setup provided in [15]. For our Drive&Segment, we train the teacher and student models with batches of size 32 and with a learning rate of $2e{-}4$ with a polynomial schedule on a single V100 GPU. During training, we perform data augmentation consisting of random image resizing in the $(0.5, 2.0)$ range, random cropping with the crop size of 512×512 pixels, random horizontal flipping, and photometric distortions.

Evaluation protocol. *Mapping.* To evaluate our models in the unsupervised setup, we run trained models on every image, thus getting segmentation predictions with values from 1 to k. Then, we compute the confusion matrix between the C ground-truth classes of the target dataset and the $k \geq C$ pseudo-classes. We map the C ground-truth classes to C out of the k pseudo-classes using Hungarian matching [32] that minimizes the overall confusion. Finally, we compute the mean IoU and pixel accuracy based on this mapping. The pixel predictions for the $k - C$ unmapped pseudo-classes are considered as false negatives.

Test datasets. We evaluate the trained models on Cityscapes [16], Dark Zurich [42], Nighttime driving [17] and ACDC [43] datasets in this fully unsupervised setup[5] *without any finetuning*. Cityscapes [16] is a well-established dataset with 500 validation images that we use for evaluation. Dark Zurich [42] and Nighttime driving [17] are two nighttime datasets, each with 50 validation images annotated for semantic segmentation that we use for evaluation. ACDC [43] is a recent dataset providing four different adverse weather conditions with 400 training and 100 validation samples per weather condition. We test our approach on the validation images annotated for semantic segmentation. The Cityscapes dataset defines 30 different semantic classes for the pixel-wise semantic segmentation task. We follow prior work, and, unless stated otherwise, we evaluate our approach on the pre-defined subset of 19 classes [16]. The same set of 19 classes is used for all other datasets.

Metrics. We evaluate results with two standard metrics for the semantic segmentation task, the mean Intersection over Union, *mIoU*, and the pixel accuracy, *PA*, as done in prior work [15]. The mIoU is the mean intersection over union averaged over all classes, while PA defines the percentage of pixels in the image that are segmented correctly, averaged over all images.

4.2 Comparison to state of the art

Here we evaluate our trained models in the unsupervised setup using the evaluation protocol described in Section 4.1. We compare our method using both the ResNet18+FPN and Segmenter [44] models to three recent unsupervised segmentation models: IIC [30], modified version of DeepCluster [9] (DC), and PiCIE [15]. In the appendix, we assess

[5] These adverse weather datasets [17,42,43] are commonly used by domain adaptation approaches that leverage unlabeled images of this type for adaptation. Here, we consider them only for evaluation to assess the generalization of our strategy; we do not have access to any of those images during training.

Table 1. Comparison to the state of the art for unsupervised semantic segmentation on Cityscapes [16] (CS), DarkZurich [42] (DZ) and Nighttime driving [17] (ND) datasets measured by the mean IoU (mIoU). The colored differences are reported with respect to the state-of-the-art approach of [15] denoted by ⚓. The *sup. init.* abbreviation stands for supervised initialization of the *encoder*, and the column *train. data* indicates whether Cityscapes (CS) or Waymo Open (WO) dataset was used for training. Please see the appendix for pixel accuracy and for results using the nuScenes dataset for training.

architecture, method	sup. init.	train. data	CS19 [16] mIoU	CS27 [16] mIoU	DZ [42] mIoU	ND [17] mIoU
RN18+FPN						
IIC† [30]	yes	CS	-	6.4 (−4.8)	-	-
Modified DC‡ [9]	yes	CS	11.3 (−4.5)	7.9 (−3.3)	7.5 (+2.9)	8.2 (−1.3)
⚓ PiCIE‡ [15]	yes	CS	15.8	11.2	4.6	9.5
Modified DC*	yes	WO	11.4 (−4.4)	7.0 (−4.1)	5.9 (+1.3)	8.2 (−1.3)
PiCIE*	yes	WO	13.7 (−2.1)	9.7 (−1.5)	4.9 (+0.3)	9.3 (−0.2)
Drive&Segment (Ours, S)	yes	WO	19.5 (+3.7)	**16.2 (+5.1)**	10.9 (+6.3)	14.4 (+4.9)
Segmenter, ViT-S/16						
Drive&Segment (Ours, S)	no	WO	**21.8 (+6.0)**	15.3 (+4.1)	**14.2 (+9.6)**	**18.9 (+9.3)**

† Results reported in [15]. ‡ Models provided by the PiCIE [15] authors.
* Trained by PiCIE code base.

Table 2. Comparison to the state-of-the-art for unsupervised semantic segmentation on the ACDC [43] dataset. Please refer to Table 1 for the used symbols. Please see the appendix for pixel accuracy and for results using the nuScenes dataset for training.

architecture, method	sup. init.	train. data	night mIoU	fog mIoU	rain mIoU	snow mIoU	average mIoU
RN18+FPN							
modified DC‡ [9]	yes	CS	8.1 (+3.7)	8.3 (−4.0)	6.9 (−5.6)	7.4 (−4.7)	7.7 (−2.6)
⚓ PiCIE‡ [15]	yes	CS	4.4	12.2	12.5	12.1	10.3
modified DC*	yes	WO	5.9 (+1.5)	11.7 (−0.5)	9.6 (−2.9)	9.8 (−2.3)	9.2 (−1.0)
PiCIE*	yes	WO	4.7 (+0.3)	14.4 (+2.1)	13.7 (+1.2)	14.3 (+2.2)	11.7 (+1.5)
Drive&Segment (Ours, S)	yes	WO	11.2 (+6.8)	14.5 (+2.3)	14.9 (+2.5)	14.6 (+2.6)	13.8 (+3.5)
Segmenter, ViT-S/16							
Drive&Segment (Ours, S)	no	WO	**13.8 (+9.4)**	**18.1 (+5.9)**	**16.4 (+3.9)**	**18.7 (+6.6)**	**16.7 (+6.5)**

the utility of the features learned by our model in other settings, such as linear probing for semantic segmentation (Sec. D), and k-NN evaluation (Sec. B.1).

We provide results on the Cityscapes, Dark Zurich, and Nighttime Driving datasets in Table 1, and show qualitative results in Figure 6. As shown in the first two columns of Table 1, our approach outperforms [15] on the Cityscapes dataset by a large margin in both the 19-class and 27-class set-ups. Improvements are visible for both architectures, but in most cases, the best results are obtained with the distilled Segmenter architecture using the ViT-S/16 backbone. The last two columns in Table 1 show results for the two nighttime segmentation datasets: Dark Zurich [42] and Nighttime Driving [17]. Our models again outperform [15] in all setups. In addition, we observe a better performance of our models compared to [15] when evaluating on the nighttime scenes. For example, on the Dark Zurich [42] dataset, the mIoU of PiCIE [15] decreases by 71% compared to the results on Cityscapes (15.8 → 4.6), while the mIoU of our Segmenter-based model decreases only by 35% (21.8 → 14.2). This suggests that our models generalize significantly better to out-of-distribution scenes. These findings hold for PiCIE models trained on both Cityscapes and on Waymo Open Dataset.

Table 3. Ablations on the Cityscapes dataset. (a) Benefits of our segment extraction method over segment proposals from [20]. **(b)** Benefits of our distillation approach showing an improvement of the student (S) over the the teacher (T) and benefits of our LiDAR cross-modal spatial constraints (LiD). **(c)** Ablation of different feature extractors for the k-means clustering.

(a) Segment extraction

arch. seg. prop.	mIoU	PA
RN18+FPN		
FH [20]	15.5	52.8
Ours	**17.4** (+1.9)	**55.9** (+3.1)
Segmenter		
FH [20]	15.8	51.8
Ours	**20.4** (+4.6)	**65.4** (+13.6)

(b) Distillation

model	LiD.	mIoU	PA
RN18+FPN			
PiCIE (T)		13.7	48.6
PiCIE (S)		14.8 (+1.1)	64.1 (+15.5)
PiCIE (S)	✓	15.1 (+1.4)	**68.4** (+19.8)
Ours (T)		17.4	55.9
Ours (S)		18.8 (+1.4)	63.4 (+7.5)
Ours (S)	✓	**19.5** (+2.1)	**66.4** (+10.5)
Segmenter			
Ours (T)		20.4	65.4
Ours (S)		20.8 (+0.4)	68.5 (+3.1)
Ours (S)	✓	**21.8** (+1.4)	**69.5** (+4.1)

(c) Feature extractors

arch. method	mIoU	PA
ViT-S/16		
DeiT [47]	21.7	73.0
DINO [10]	20.2	64.4
ResNet18		
supervised [26]	19.6	70.0
ResNet50		
supervised [26]	21.3	67.6
OBOW [22]	20.7	65.9
PixPro [53]	20.7	65.9
MaskCon. [48]	19.1	68.0

Finally, Table 2 shows results on the ACDC dataset in four different adverse weather conditions. Results follow a similar trend as in Table 1 and show the superiority of our approach measured by mIoU compared to the current state-of-the-art unsupervised semantic segmentation method of [15] on images out of the training distribution, such as images at night or in snow. Please see the appendix for the complete set of results, including results using nuScenes (Sec. A.1), pixel accuracy (Sec. A.4), per-class results (Sec. A.5) and analysis of the confusion matrices (Sec. B.3).

4.3 Ablations

In this section, we ablate the main components of our approach. In particular, we study the benefits of our cross-modal segment extraction (Table 3a), of our distillation with cross-modal spatial constraints (Table 3b), effect of varying feature extractors for the k-means clustering (Table 3c), variance of the results over multiple runs, the influence of LiDAR resolution and the number of clusters used in k-means.

Segment extraction approach. To evaluate the benefits of our cross-modal segment extraction module, we investigate using segment proposals generated with a purely image-based segmentation approach by Felzenszwalb and Huttenlocher (FH) [20]. It groups pixels into segments based on similar color and texture properties. We use the same set of hyperparameters as [27]. The results are shown in Table 3a and demonstrate clear benefits of our LiDAR-based cross-modal segment extraction method despite the difficulties of using LiDAR data discussed in Section 3.1. We attribute the better results of our approach to the fact that LiDAR data segmentation operates with range information, which is much stronger at separating objects from the background and from each other compared to the purely image-based approach of FH [20]. Indeed, FH relies only on color/texture and is therefore much more likely to join multiple objects into one segment or separate a single object into multiple segments. The benefits of our cross-modal segment extraction are observed for both studied architectures.

Distillation with cross-modal spatial constraints. To evaluate the benefits of our teacher-student distillation method with cross-modal spatial constraints (Section 3.3), we compare the predictions of the teacher T (before distillation) and the student S (after distillation). Table 3b presents results on the Cityscapes dataset using both convolutional- and transformer-based architectures. The results show consistent improvements using our distillation technique, particularly regarding the pixel accuracy metric. We believe that this could be attributed to improvements in predictions for classes such as vegetation and buildings. They often occupy large areas of the image and benefit most from the distillation as they are usually not well covered by the LiDAR scans. Furthermore, the results show clear benefits of using this distillation step both with and without cross-modal spatial constraints (LiD) by Student S outperforming Teacher T in both scenarios. Please also note that our distillation technique works well even in combination with another training approach (PiCIE [15]).

Sensitivity to the initialization. To study the influence of initialization, we take the features extracted by DINO [10] and run the k-means clustering (Section 3.2) four times. For each of the k-means clustering outcomes, we run the segmentation model training four times with different initializations. The variance over all k-means and training runs is only 0.5 for mIoU and 1.5 for pixel accuracy (i.e., $20.4 \pm 0.5/65.4 \pm 1.5$). These results clearly show that our method is not very sensitive to k-means initialization or to the network initialization.

Feature extractors. An ablation of seven different feature extractors (convolutional- and Transformer-based) for the task of segment-wise unsupervised labelling is shown in Table 3c. The results on the Cityscapes [16] dataset using our Segmenter model demonstrate that our approach works well with several different feature extractors.

LiDAR resolution and number of clusters. An ablation of the influence of LiDAR resolution is shown in Sec. A.3 in the appendix and demonstrates that our method is robust to LiDAR's sparsity. Furthermore, we study the choice of the number of clusters for the k-means clustering in Sec. A.2 in the appendix.

4.4 Limitations and failure modes

Our approach has the following three main limitations. First, LiDAR point clouds do not provide information about very distant or even infinitely distant objects, e.g., the sky, which our approach cannot learn to segment. Second, LiDAR point clouds paired with geometric segmentation can not correctly distinguish road from sidewalk or grass, when all surfaces are similarly flat. Both the above limitations might be possibly tackled by pairing our LiDAR-based segment proposals with an unsupervised image-based method such as [20]. Also, the LiDAR points must not be too sparse (e.g., only 4 beams), since otherwise the LiDAR-based segments would be of poor quality. However, this is not an overly restricting requirement as it is common to use LiDAR sensors with sufficient beam resolution, e.g., as in the recent Waymo Open [45] or ONCE [35] datasets. Finally, we encounter semantically similar objects appearing in multiple pseudo-classes, a natural side effect of clustering. This issue may be mitigated by using different feature clustering methods, e.g., from the family of graph clustering methods, that allow the measurement of similarities on manifolds in the feature space instead of the currently used Euclidean metric in the k-means clustering.

5 Conclusion

We have developed Drive&Segment- a fully unsupervised approach for semantic image segmentation in urban scenes. The approach relies on novel modules for (i) cross-modal segment extraction and (ii) distillation with cross-modal constraints that leverage LiDAR point clouds aligned with images. We evaluate our approach on four different autonomous driving datasets in challenging weather and illumination conditions and demonstrate major gains over prior work. This work opens up the possibility of large-scale autonomous learning of embodied perception models without explicit human supervision.

Acknowledgments. This work was partly supported by the European Regional Development Fund under the project IMPACT (reg. no. CZ.02.1.010.00.015_0030000468), and the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90140). Antonin Vobecky was partly supported by the CTU Student Grant Agency (reg. no. SGS21184OHK33T37).

References

1. Afouras, T., Asano, Y.M., Fagan, F., Vedaldi, A., Metze, F.: Self-supervised object detection from audio-visual correspondence. In: arXiv (2021) 4
2. Alayrac, J.B., Recasens, A., Schneider, R., Arandjelovic, R., Ramapuram, J., De Fauw, J., Smaira, L., Dieleman, S., Zisserman, A.: Self-supervised multimodal versatile networks. In: NeurIPS (2020) 4
3. Alwassel, H., Mahajan, D., Korbar, B., Torresani, L., Ghanem, B., Tran, D.: Self-supervised learning by cross-modal audio-video clustering. In: NeurIPS (2020) 4
4. Arandjelovic, R., Zisserman, A.: Look, listen and learn. In: ICCV (2017) 4
5. Bartoccioni, F., Zablocki, É., Pérez, P., Cord, M., Alahari, K.: Lidartouch: Monocular metric depth estimation with a few-beam lidar. In: arXiv (2021) 4
6. Bielski, A., Favaro, P.: Emergence of object segmentation in perturbed generative models. In: NeurIPS (2019) 4
7. Bogoslavskyi, I., Stachniss, C.: Efficient online segmentation for sparse 3d laser scans. PFG (2017) 5
8. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR (2020) 3, 10, 18
9. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: ECCV (2018) 4, 8, 9, 10, 11, 18, 19, 20
10. Caron, M., Touvron, H., Misra, I., Jegou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging Properties in Self-Supervised Vision Transformers. In: ICCV (2021) 6, 8, 9, 12, 13
11. Chen, H., Xie, W., Afouras, T., Nagrani, A., Vedaldi, A., Zisserman, A.: Localizing visual sounds the hard way. In: CVPR (2021) 4
12. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018) 3
13. Chen, M., Artières, T., Denoyer, L.: Unsupervised object segmentation by redrawing. In: NeurIPS (2019) 4

14. Cheng, B., Collins, M.D., Zhu, Y., Liu, T., Huang, T.S., Adam, H., Chen, L.C.: Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In: CVPR (2020) 3
15. Cho, J.H., Mall, U., Bala, K., Hariharan, B.: PiCIE: Unsupervised semantic segmentation using invariance and equivariance in clustering. In: CVPR (2021) 1, 4, 8, 9, 10, 11, 12, 13, 18, 19, 20, 21, 22, 23, 24, 25, 26
16. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016) 3, 9, 10, 11, 13, 18, 20, 21, 24, 25, 26
17. Dai, D., Van Gool, L.: Dark model adaptation: Semantic image segmentation from daytime to nighttime. In: IEEE ITSC (2018) 3, 10, 11, 18, 20
18. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR (2009) 3, 8, 9, 26
19. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021) 3, 6, 8
20. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV (2004) 12, 13
21. French, G., Laine, S., Aila, T., Mackiewicz, M., Finlayson, G.: Semi-supervised semantic segmentation needs strong, varied perturbations. BMVC (2020) 21, 22, 25, 26
22. Gidaris, S., Bursuc, A., Puy, G., Komodakis, N., Cord, M., Pérez, P.: Obow: Online bag-of-visual-words generation for self-supervised learning. In: CVPR (2021) 4, 12
23. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: ICLR (2018) 4
24. Grill, J., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.Á., Guo, Z., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M.: Bootstrap your own latent - A new approach to self-supervised learning. In: NeurIPS (2020) 4
25. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.B.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020) 4
26. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 8, 12
27. Hénaff, O.J., Koppula, S., Alayrac, J.B., Oord, A.v.d., Vinyals, O., Carreira, J.: Efficient visual pretraining with contrastive detection. In: ICCV (2021) 4, 12
28. Hwang, J.J., Yu, S.X., Shi, J., Collins, M.D., Yang, T.J., Zhang, X., Chen, L.C.: Segsort: Segmentation by discriminative sorting of segments. In: ICCV. pp. 7334–7344 (2019) 4
29. Jaritz, M., Vu, T.H., Charette, R.d., Wirbel, E., Pérez, P.: xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation. In: CVPR (2020) 4
30. Ji, X., Henriques, J.F., Vedaldi, A.: Invariant information clustering for unsupervised image classification and segmentation. In: ICCV (2019) 4, 8, 9, 10, 11, 20, 24, 25
31. Kanezaki, A.: Unsupervised image segmentation by backpropagation. In: ICASSP (2018) 4
32. Kuhn, H.W., Yaw, B.: The hungarian method for the assignment problem. NRLQ (1955) 10
33. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017) 3, 8
34. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015) 3
35. Mao, J., Niu, M., Jiang, C., Liang, H., Liang, X., Li, Y., Ye, C., Zhang, W., Li, Z., Yu, J., et al.: One million scenes for autonomous driving: Once dataset. NeurIPS (2021) 13
36. Miech, A., Alayrac, J.B., Smaira, L., Laptev, I., Sivic, J., Zisserman, A.: End-to-end learning of visual representations from uncurated instructional videos. In: CVPR (2020) 4
37. Neuhold, G., Ollmann, T., Rota Bulo, S., Kontschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. In: ICCV (2017) 3

38. Ouali, Y., Hudelot, C., Tami, M.: Autoregressive unsupervised image segmentation. In: ECCV (2020) 4
39. Owens, A., Efros, A.A.: Audio-visual scene analysis with self-supervised multisensory features. In: ECCV (2018) 4
40. Recasens, A., Luc, P., Alayrac, J.B., Wang, L., Strub, F., Tallec, C., Malinowski, M., Pătrăucean, V., Altché, F., Valko, M., Grill, J.B., van den Oord, A., Zisserman, A.: Broaden your views for self-supervised video learning. In: ICCV (2021) 4
41. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015) 3
42. Sakaridis, C., Dai, D., Van Gool, L.: Map-guided curriculum domain adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. IEEE TPAMI (2020) 3, 10, 11, 18, 20
43. Sakaridis, C., Dai, D., Van Gool, L.: ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In: ICCV (2021) 3, 9, 10, 11, 19, 20, 24, 29
44. Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: ICCV (2021) 3, 8, 10
45. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR (2020) 3, 9, 10, 13, 21, 23, 24
46. Tian, H., Chen, Y., Dai, J., Zhang, Z., Zhu, X.: Unsupervised object detection with lidar clues. In: CVPR (2021) 4
47. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: ICML (2021) 12
48. Van Gansbeke, W., Vandenhende, S., Georgoulis, S., Van Gool, L.: Unsupervised semantic segmentation by contrasting object mask proposals. In: ICCV (2021) 1, 4, 12
49. Varma, G., Subramanian, A., Namboodiri, A., Chandraker, M., Jawahar, C.: Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In: WACV (2019) 3
50. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al.: Deep high-resolution representation learning for visual recognition. IEEE TPAMI (2020) 3
51. Weston, R., Cen, S., Newman, P., Posner, I.: Probably unknown: Deep inverse sensor modelling radar. In: ICRA (2019) 4
52. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. In: NeurIPS (2021) 3
53. Xie, Z., Lin, Y., Zhang, Z., Cao, Y., Lin, S., Hu, H.: Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In: CVPR (2021) 12
54. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: CVPR (2020) 3
55. Yuan, Y., Chen, X., Wang, J.: Object-contextual representations for semantic segmentation. In: ECCV (2020) 3
56. Zamir, A.R., Sax, A., Shen, W., Guibas, L.J., Malik, J., Savarese, S.: Taskonomy: Disentangling task transfer learning. In: CVPR (2018) 3
57. Zhang, X., Maire, M.: Self-supervised visual representation learning from hierarchical grouping. In: NeurIPS (2020) 4
58. Zhao, H., Gan, C., Rouditchenko, A., Vondrick, C., McDermott, J., Torralba, A.: The sound of pixels. In: ECCV (2018) 4
59. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR (2017) 3

60. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., Zhang, L.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: CVPR (2021) 3

Appendix

Table of Contents

A Additional quantitative results

A.1 Results with another training dataset

We report in Tables 4 and 5 the performance of Drive&Segment when trained using a subset of ∼8k images from the nuScenes dataset [8]. As shown in Table 4, the mIoU on Cityscapes is 19.8. Although there is a small drop from the 21.8 achieved with Drive&Segment trained on Waymo Open, the results are still significantly better than those of the competing methods. This drop might be caused by differences of statistics between the two datasets, *e.g.*, nuScenes has fewer examples of smaller-object classes, such as pedestrians.

Table 4. Comparative results of unsupervised semantic segmentation methods when trained on nuScenes. Comparison to the state of the art on Cityscapes [16] (CS), DarkZurich [42] (DZ) and Nighttime Driving [17] (ND) datasets measured by the mean IoU (mIoU). The colored differences are reported with respect to the state-of-the-art approach of [15] denoted by ⚓; '*sup. init.*' stands for supervised initialization of the *encoder* and the column '*train. data*' indicates the dataset used for training, namely nuScenes [8] (nuSC).

architecture, method	sup. init.	train. data	CS19 [16] mIoU	CS27 [16] mIoU	DZ [42] mIoU	ND [17] mIoU
RN18+FPN						
⚓ PiCIE* [15]	yes	nuSC	15.8	9.7	4.6	9.9
Modified DC* [9]	yes	nuSC	11.6 (−4.2)	7.1 (−2.6)	7.7 (+3.1)	8.3 (−1.6)
Drive&Segment (Ours, S)	yes	nuSC	16.2 (+0.4)	11.4 (+1.7)	7.5 (+2.9)	10.2 (+0.3)
Segmenter, ViT-S/16						
Drive&Segment (Ours, S)	no	nuSC	19.8 (+4.0)	13.9 (+4.2)	9.7 (+5.1)	14.1 (+4.2)

* Our training using PiCIE code base.

Table 5. Comparative results on ACDC when methods trained on nuScenes. Comparison to the state of the art for unsupervised semantic segmentation on the ACDC [43] dataset. Please refer to Table 4 for the symbols.

architecture, method	sup. init.	train. data	night mIoU	fog mIoU	rain mIoU	snow mIoU	average mIoU
RN18+FPN							
⚓ PiCIE* [15]	yes	nuSC	4.3	8.9	9.5	7.5	7.5
Modified DC* [9]	yes	nuSC	6.7 (+2.4)	11.7 (+2.8)	10.4 (+0.9)	9.6 (+2.1)	9.6 (+2.1)
Drive&Segment (Ours, S)	yes	nuSC	7.9 (+3.6)	14.3 (+5.4)	14.4 (+4.9)	13.4 (+5.9)	12.5 (+5.0)
Segmenter, ViT-S/16							
Drive&Segment (Ours, S)	no	nuSC	10.6 (+6.3)	13.3 (+4.4)	16.0 (+6.5)	14.8 (+7.3)	13.9 (+6.4)

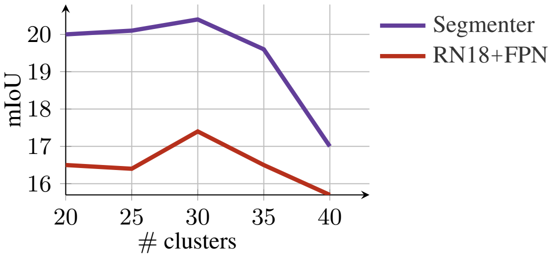


Fig. 7. Ablation of the number of clusters. Performance in mIoU, when using the **Segmenter** model and the **ResNet18+FPN** model on the Cityscapes dataset, as a function of the number of clusters in the unsupervised labeling step.

A.2 Ablation of the number of clusters in unsupervised labeling

Here we investigate the sensitivity of our method to the number k of clusters used for unsupervised labeling. Figure 7 shows the mIoU results on Cityscapes for $k \in \{20, 25, 30, 35, 40\}$. In all cases we use a ViT-S/16 feature extractor trained with DINO. The results show that for $k \in \{20, 25, 30, 35\}$ the *mIoU performance is fairly stable*. As expected, when the number of clusters becomes much higher than the number of Cityscapes classes (*e.g.*, $k = 40$), the performance drops.

A.3 Influence of the LiDAR's density

We investigate here the performance of Drive&Segment when provided with sparser LiDAR data. We performed experiments on the Waymo Open dataset and downsampled the LiDAR data from 64 to 32 beam channels by dropping every other channel. We re-trained the *teacher* model three times and report the average performance (following the setup of the main paper). We obtained 20.3 mIoU, which is only slightly lower than the 20.4 obtained with the full LiDAR resolution, demonstrating the robustness of our method to this considerable decrease of LiDAR resolution. However, as already discussed in the main paper, our method will likely not work well with extremely sparse LiDAR data (*e.g.*, low-cost LiDARs with 4-beam channels). Such a sparsity would lead to poor LiDAR-based segments and geometric priors that would rather confuse the model, instead of teaching it to recognize objects.

Table 6. Comparative results using PA metric. Comparison to the state of the art for unsupervised semantic segmentation on Cityscapes [16] (CS), DarkZurich [42] (DZ) and Nighttime driving [17] (ND) datasets measured by the pixel accuracy (PA). Same organization as Table 4. For easy reference, rows are colored according to the used training dataset.

architecture, method	sup. init.	train. data	CS19 [16] PA	CS27 [16] PA	DZ [42] PA	ND [17] PA
RN18+FPN						
⚓ PiCIE‡ [15]	yes	CS	63.1	62.7	30.7	41.4
IIC† [30]	yes	CS	-	47.9 (−14.8)	-	-
Modified DC‡ [9]	yes	CS	52.4 (−10.7)	52.1 (−10.7)	42.4 (+11.7)	46.2 (+4.8)
Modified DC*	yes	nuSc	45.9 (−17.2)	45.7 (−17.0)	41.4 (+10.7)	41.9 (+0.5)
PiCIE*	yes	nuSc	61.6 (−1.5)	61.3 (−1.4)	29.6 (−1.1)	45.1 (+3.7)
Drive&Segment (Ours, S)	yes	nuSc	61.4 (−1.7)	61.1 (−1.6)	37.4 (+6.7)	33.6 (−7.8)
Modified DC*	yes	WO	55.6 (−7.5)	43.2 (−19.5)	35.8 (+5.1)	33.4 (−8.0)
PiCIE*	yes	WO	48.6 (−14.5)	48.3 (−14.4)	31.9 (+1.1)	40.0 (−1.4)
Drive&Segment (Ours, S)	yes	WO	66.4 (+3.3)	67.1 (+4.3)	47.7 (+17.0)	49.0 (+7.6)
Segmenter, ViT-S/16						
Drive&Segment (Ours, S)	no	nuSc	73.2 (**+10.1**)	72.8 (**+10.1**)	50.2 (+19.5)	65.5 (**+24.1**)
Drive&Segment (Ours, S)	no	WO	**69.5** (+6.4)	**69.1** (+6.4)	**55.9** (+25.1)	**60.2** (+18.8)

† Results reported in [15]. ‡ Models provided by the PiCIE [15] authors.
* Trained by PiCIE code base.

Table 7. Comparative results on ACDC using PA metric. Comparison to the state-of-the-art approach [15] for unsupervised semantic segmentation on the ACDC [43] dataset. Same organization as Table 5. For easy reference, rows are colored according to the used training dataset

method	sup. init.	train. data	night PA	fog PA	rain PA	snow PA	average PA
RN18+FPN							
⚓ PiCIE [15]	yes	CS	25.8	50.0	53.6	50.4	45.0
MDC [9]	yes	CS	43.0 (+17.3)	43.6 (−6.4)	35.0 (−18.6)	38.8 (−11.5)	40.1 (−4.8)
Modified DC*	yes	nuSC	36.5 (+10.7)	44.8 (−5.2)	41.4 (−12.2)	38.5 (−11.9)	40.3 (−4.7)
PiCIE*	yes	nuSC	26.9 (+1.1)	33.1 (−16.9)	33.4 (−20.2)	29.1 (−21.3)	30.6 (−14.4)
Drive&Segment (Ours, S)	yes	nuSC	34.5 (+8.7)	59.4 (+9.4)	58.2 (+4.6)	53.9 (+3.5)	51.5 (+6.5)
MDC*	yes	WO	32.9 (+7.2)	47.0 (−3.0)	40.3 (−13.3)	44.2 (−6.2)	41.1 (−3.8)
PiCIE*	yes	WO	27.2 (+1.4)	56.9 (+6.8)	53.8 (+0.2)	53.0 (+2.6)	47.7 (+2.8)
Drive&Segment (Ours, S)	yes	WO	43.2 (+17.5)	56.5 (+6.5)	54.1 (+0.5)	55.5 (+5.1)	52.3 (+7.4)
Segmenter, ViT-S/16							
Drive&Segment (Ours, S)	no	nuSC	50.2 (+24.4)	60.2 (+10.2)	62.5 (+8.9)	56.5 (+6.1)	57.5 (+12.5)
Drive&Segment (Ours, S)	no	WO	**52.6** (+26.9)	54.2 (+4.2)	50.1 (−3.5)	**56.8** (+6.4)	**53.4** (+8.5)

A.4 Pixel accuracy results

In Tables 6 and 7, we report results measured with the pixel accuracy (PA) metric corresponding to all experiments of our main paper. We observe that results follow a similar trend to those measured with mIoU.

A.5 Category-wise results

In the main paper, we have presented results averaged over all classes. We report in Table 8 the *per-class IoU* results of our Drive&Segment approach on the Cityscapes dataset.

We observe that Drive&Segment outperforms the baseline PiCIE on 15 out of 19 classes. IoU gains (w.r.t. PiCIE trained on Waymo Open dataset) are significant for

Table 8. Per-class comparative performance on Cityscapes. Per-class IoU is evaluated using the Hungarian algorithm on the 19 validation classes. We can see significant benefits of Drive&Segment ('D&S') over PiCIE in 14 (including all road users and objects) out of 19 classes. Drive&Segment works much worse for *sidewalk* and *sky* as we discuss in Sections A.5 and B.4. '`(CS)`' stands for a model trained on the Cityscapes [16] dataset, while '`(WO)`' for models trained on the Waymo Open [45] dataset. The best results per class are highlighted in bold and color.

	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mIoU
RN18+FPN																				
PiCIE [15] `(CS)`	58.2	12.5	63.8	1.0	2.4	1.3	0.1	0.4	55.5	1.7	44.7	1.9	0.5	48.2	1.3	3.9	1.0	0.5	1.6	15.8
PiCIE [15] `(WO)`	58.5	**13.8**	35.8	**6.7**	0.7	1.2	0.4	1.2	28.3	1.2	**55.8**	3.1	0.6	48.5	0.5	1.5	0.3	0.0	2.3	13.7
D&S (Ours, `WO`)	72.7	7.0	56.6	4.5	**5.6**	16.9	3.6	15.7	**66.8**	**2.2**	6.0	40.0	**5.0**	44.7	0.5	18.5	0.2	**1.4**	2.1	19.5
Segmenter, ViT-S/16																				
D&S (Ours, `WO`)	**74.1**	7.0	**65.7**	**6.6**	1.0	**24.9**	**4.3**	**16.6**	64.8	1.8	3.7	**45.9**	4.3	**57.3**	**1.7**	**19.9**	**1.3**	0.4	**12.1**	**21.8**

small-object classes such as *pole* (+23.2/+15.2 with Segmenter and ResNet18+FPN respectively), *traffic signs* (+15.4/+14.5), and *person* (+42.8/+36.9). They are also substantial for some classes that can cover larger image portions, *e.g.*, *road* (+15.6/+14.2), *vegetation* (+36.5/+38.5), *car* (+8.8/−3.8). The results of ResNet18+FPN are slightly worse on the *car* class because *car* instances are split into several pseudo-classes. Gains over *road* and *car* were expected since LiDAR data provide very good segments for these classes; it is more surprising to see gains on *vegetation*, a class that is not easily captured by LiDAR.

B Analyzing learned representations

B.1 k-NN evaluation of learned representations

To evaluate the quality of the learned representations, we compare the representations produced by a ResNet18 backbone trained (a) on Imagenet in a fully-supervised fashion for the classification task, (b) using PiCIE [15] trained on Waymo Open, and (c) using our Drive&Segment trained on Waymo Open. For this comparison we perform k-NN based pixel-wise classification on the Cityscapes validation set using a *low-shot scenario* where only 100 Cityscapes training images are available (we consider three random splits of 100 images from [21] and report the average results). Our goal is to analyze the ability of the representations to learn with few training examples. In Table 9, we report results in terms of pixel accuracy for $k \in \{1, 5, 20\}$ and observe that Drive&Segment outperforms both the supervised baseline and PiCIE [15].

B.2 Representation analysis via PCA

In Figure 8, we visualize the three main PCA components of the *decoder* features as RGB. We observe that our features learned with Segmenter separate better object classes.

Table 9. Evaluation of learned features using k-NN pixel-wise classification. Results are produced by running k-NN with three different 100-image training sets [21] and computing the average (over the three runs) pixel accuracy on the Cityscapes validation split. Results are reported with the Pixel Accuracy (PA) metric.

method	$k = 1$	$k = 5$	$k = 20$
supervised	76.9	79.4	81.2
PiCIE [15]	74.3 (-2.6)	78.0 (-1.4)	79.1 (-2.1)
Drive&Segment	81.1 (+4.2)	83.2 (+3.8)	84.7 (+3.5)

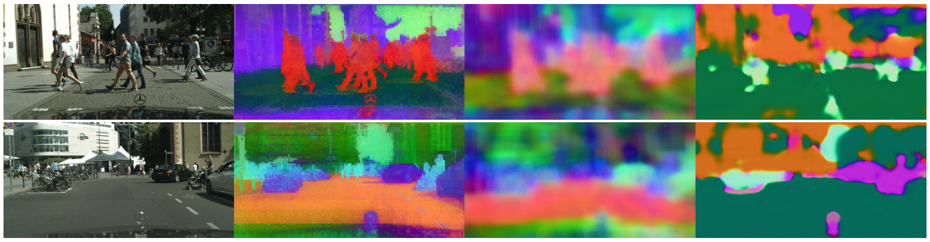
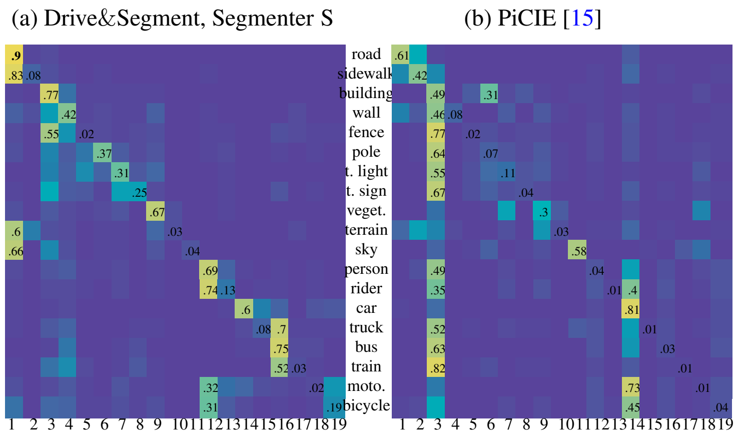


Fig. 8. Feature visualization. We do PCA analysis of the pixel-wise decoder features from each image (independenly between the different images) and visualize the three first PCA components as an RGB image. 'Segm.' stands for Segmenter with ViT-S/16 and 'RN' for ResNet18+FPN.

B.3 Confusion matrices for class mapping

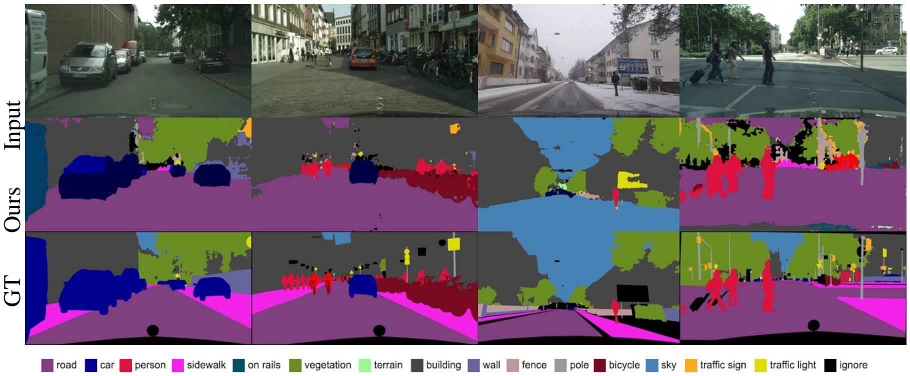
Here we analyze the confusion matrices, presented in Figure 9, which provide the mapping between ground truth and pseudo classes. For each confusion matrix, we reorder the columns based on the matching obtained from the Hungarian algorithm, and L_1-normalize the values per row, i.e., per ground-truth class (for simplicity, we do not illustrate the un-matched pseudo-classes in the figures). Thus, a value of 1 would signify that all pixels in a ground-truth class belong to a single pseudo-class. Moreover, due to the reordering, the largest values should ideally be on the diagonal of the confusion matrix.

For each row, the highest and the diagonal entry are reported. We note that, for Drive&Segment (Fig. 9(a)), 90% of the road pixels are covered by the first pseudo-class. However, this pseudo-class also covers large portions of sidewalk and vegetation as all these labels belong to ground pixels and hence are segmented together by our LiDAR-based segment proposal mechanism. Similarly, pseudo-class 12 overlaps person, rider, motorcycle and bicycle, i.e., with human-related ground-truth classes. Regarding PiCIE (Fig. 9(b)), only a few pseudo-classes have a significant overlap with ground-truth classes. In particular, the pseudo-class 3 overlaps with the majority of the ground-truth classes.



(a) Drive&Segment, Segmenter S (b) PiCIE [15]

Fig. 9. Row-normalized confusion matrices.



(a) mixed labels (b) good shape, wrong class (c) sky ↔ road confusion

Fig. 10. Failure cases. **(a)** Due to the noise in the training data (discussed in Section B.4; images and LiDAR point clouds come from the Waymo Open [45] dataset), Drive&Segment sometimes predicts multiple pseudo-labels inside the same object (here different shades of blue inside the car on the left). **(b)** Objects that belong to the same semantic category (*e.g.*, *cars*) might end-up clustered into different pseudo-classes due to differences in appearance (*e.g.*, a separate pseudo-class that corresponds to the rear of the cars). **(c)** The *road*↔*sky* misplacement/confusion is caused by the absence of sky-occupied labeled pixels at training as they are not covered by the LiDAR data. Therefore, the model assigns the most common label to the sky, which is the pseudo-label that corresponds to the road. This leads to either predicting the road as *sky* (third column), or predicting sky as *road* (fourth column), depending on the outcome of the Hungarian matching.

B.4 Failure cases

The main limitations of our Drive&Segment approach are discussed in Section 4.4 of the main paper. Here, we show some qualitative examples of these failure modes and discuss more thoroughly their roots.

The first limitation of Drive&Segment is the complete absence of pseudo-labeled training data for the *sky* class. This is because the LiDAR data do not capture the sky. As a consequence, our models learn to classify the *sky* pixels as *road* (see the “sky” row of the confusion matrix in Figure 9a), which is the most dominant (pseudo-)label in the data. We provide examples of this behavior in Figure 10(c).

The second most common failure mode is inherited from the object proposal method that relies only on geometry-derived features. Specifically, the segment proposal method might over-segment an object, potentially causing different object parts being assigned to different pseudo-labels. The class majority voting in our refinement stage does not always rectify this issue. As a consequence, our models might learn to make predictions that mix multiple pseudo-labels in one object. For example, see Figure 10(a) where the car is mixed with the *truck* class.

Finally, our LiDAR-based proposal method groups all points from the ground plane into a single segment, without being able to distinguish the various ground-plane classes (*e.g.*, *road*, *sidewalk* and *terrain*) that are defined in the image domain. Figure 10 provides examples of this failure mode. This phenomenon is also well visible in Figure 9a.

C Additional qualitative results

C.1 Qualitative comparison to previous work

We show a qualitative comparison with IIC [30] and PiCIE [15] in Figure 11. For a fair comparison, we use the same samples and the visualization protocol as in [15]. Note that these samples come from the PiCIE and IIC training set, namely from the *train* set of the Cityscapes [16] dataset, while for our method these are only test samples. In Figure 11, note how our Drive&Segment is able to segment the *person* class, while neither IIC nor PiCIE are capable to do so.

C.2 Qualitative Results

In Figures 12 and 13, we report Drive&Segment predictions on Cityscapes validation images. In spite of the domain gap between the training dataset (Waymo Open Dataset [45] with images from US cities) and the Cityscapes test set, our approach produces convincing results. Furthermore, in Figure 14, we report qualitative results of our method pretrained on *daytime-only* images and evaluated on *out-of-training-distribution* splits of ACDC [43], *e.g.*, *night*, *snow* or *fog*. We discuss the main failure modes in Section B.4.

D Evaluating Drive&Segment with supervised fine-tuning

The goal of our work is to train image segmentation models without any human annotation. Here, we evaluate with some preliminary experiments the applicability of the

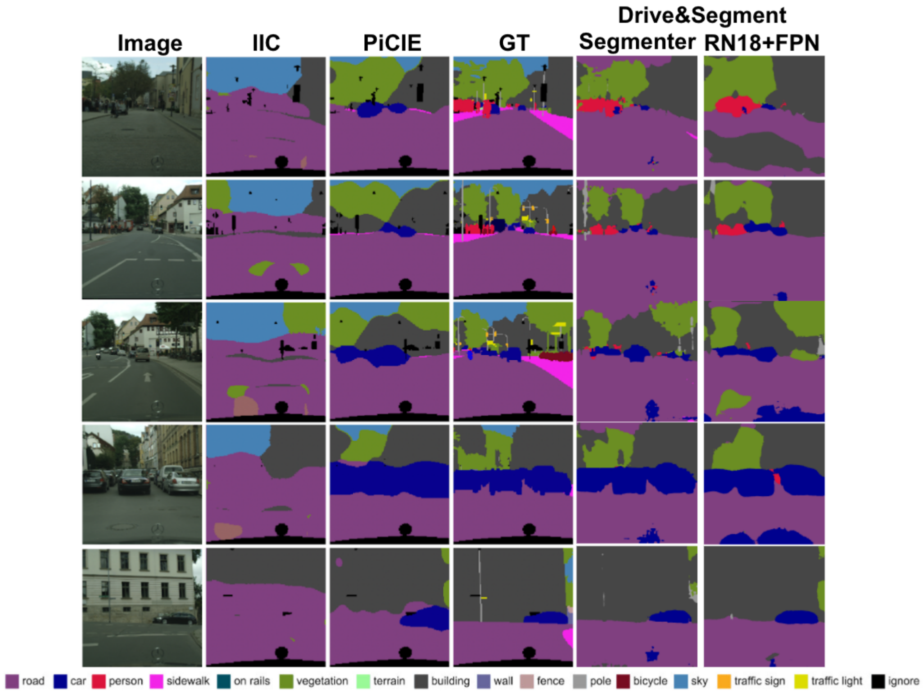


Fig. 11. Qualitative comparison of PiCIE [15], IIC [30] and our Drive&Segment approach on PiCIE *training* samples. For a fair comparison, we use the same visualization procedure as in [15]. Results are shown on center-cropped Cityscapes training images. Note that our method is able to capture objects' contours much better and to segment categories such as *person* that are not visible in IIC or PiCIE results.

Table 10. Supervised fine-tuning on the Cityscapes [16] semantic segmentation task. Results report mean Intersetion over Union (mIoU). We fine-tune the pre-trained ResNet18+FPN networks on either the entire Cityscapes training split ('Full Cityscapes') or only 100 images from the training split ('Low-shot') [21] and test on the Cityscapes validation split. 'Linear' fine-tunes only the last linear layer, 'Decoder+Linear' fine-tunes the FPN decoder and the last linear layer, and 'End-to-End' fine-tunes the entire network.

	Full Cityscapes		Low-shot
Pre-training	Linear	Decoder+Linear	End-to-End
PiCIE [15]	17.4	29.5	30.4
Imagenet (supervised)	25.7	41.9	48.2
Drive&Segment	**36.4**	**46.4**	**49.2**

proposed Drive&Segment method on a different but related task, that of *self-supervised pre-training* of semantic segmentation networks (i.e., self-supervised feature learning). Specifically, we take the ResNet18+FPN model trained with Drive&Segment, replace its last linear prediction layer with a new layer that has as many outputs as classes in Cityscapes (19), and fine-tune the resulting network on the Cityscapes [16] dataset using available human annotations. We compare against (a) using PiCIE [15] for self-supervised pre-training and (b) *supervised* pre-training on ImageNet [18].

We evaluate the different pre-training approaches with three fine-tuning setups. The first setup is to freeze both the ResNet18 backbone and the FPN decoder (i.e., keep their pre-trained weights fixed) and fine-tune only the last linear prediction layer. The second setup is to freeze only the ResNet18 backbone and fine-tune both the FPN decoder and the last linear layer. In both cases, we train on the entire training split (2975 images) of Cityscapes. The goal of these first two setups is to evaluate the quality of the pre-trained ResNet18+FPN (1st setup) or ResNet18 (2nd setup) features as they are. The third setup targets the *low-shot scenario*: the segmentation network is fine-tuned end-to-end using only 100 Cityscapes training images (we consider three random splits of 100 images from [21]). The purpose of this setup is to evaluate the strength of the pre-trained network in a regime where only a few annotations are available for fine-tuning.

In the first two setups we train for $40k$ iterations, and we train for $4k$ in the low-shot setup. In all setups, we use SGD with momentum set to 0.9, weight decay to 0.0005 and mini-batches of size 8. During training we use random image scaling (by a ratio in $[0.5, 2.0]$), random cropping (with size 769) and horizontal flipping. At test time, we use the original image size and horizontal flip augmentations. The learning rates were tuned for each fine-tuning setup and each evaluated method separately.

We report results in Table 10. Although our method was not designed or optimized for self-supervised feature pre-training, it still provides promising results that surpass both PiCIE and ImageNet pre-training.

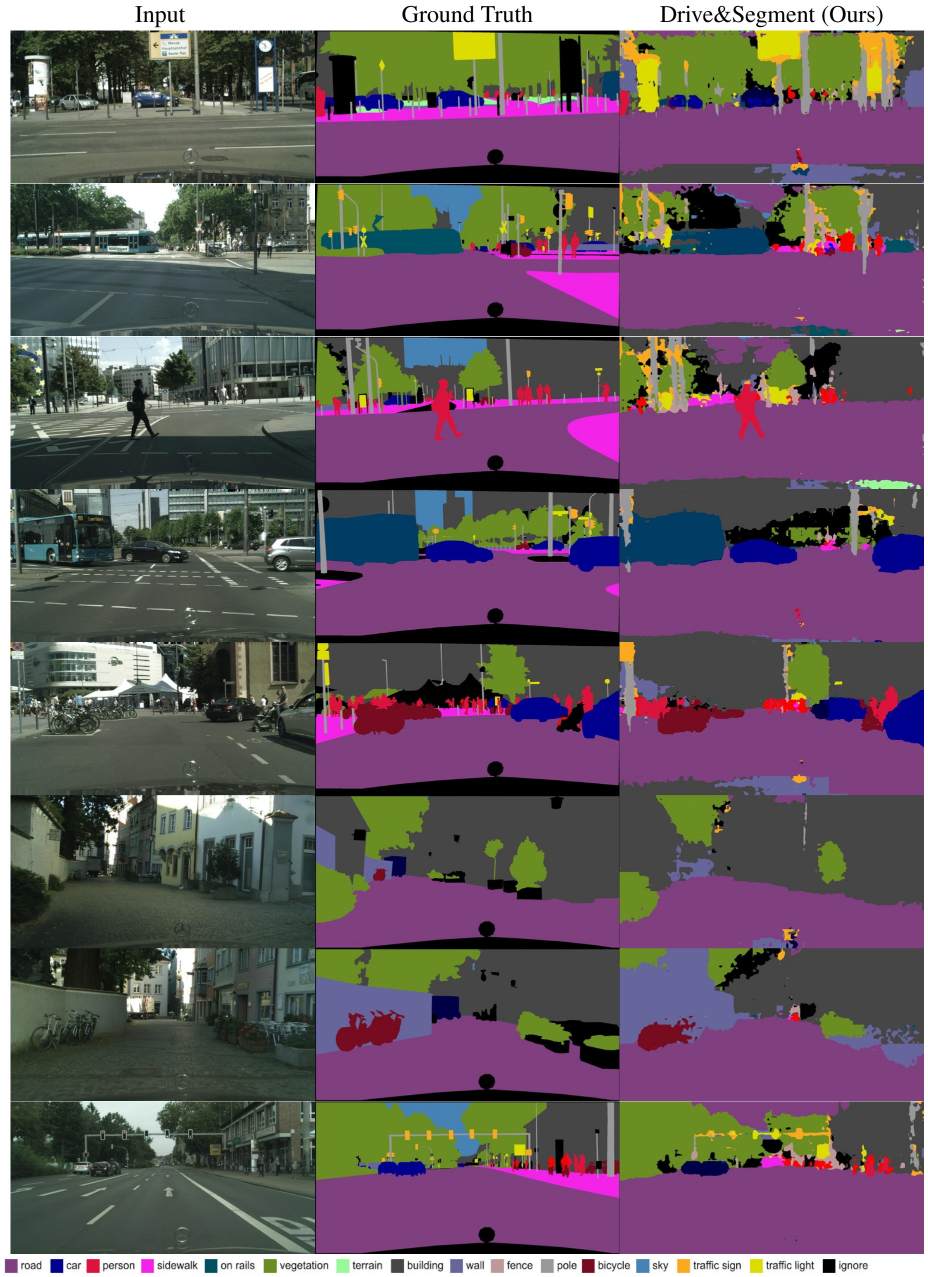


Fig. 12. Qualitative results for unsupervised semantic segmentation using our Drive&Segment approach on the validation split of the Cityscapes dataset. The matching between our pseudo-classes and the set of ground-truth classes is obtained using the Hungarian algorithm.

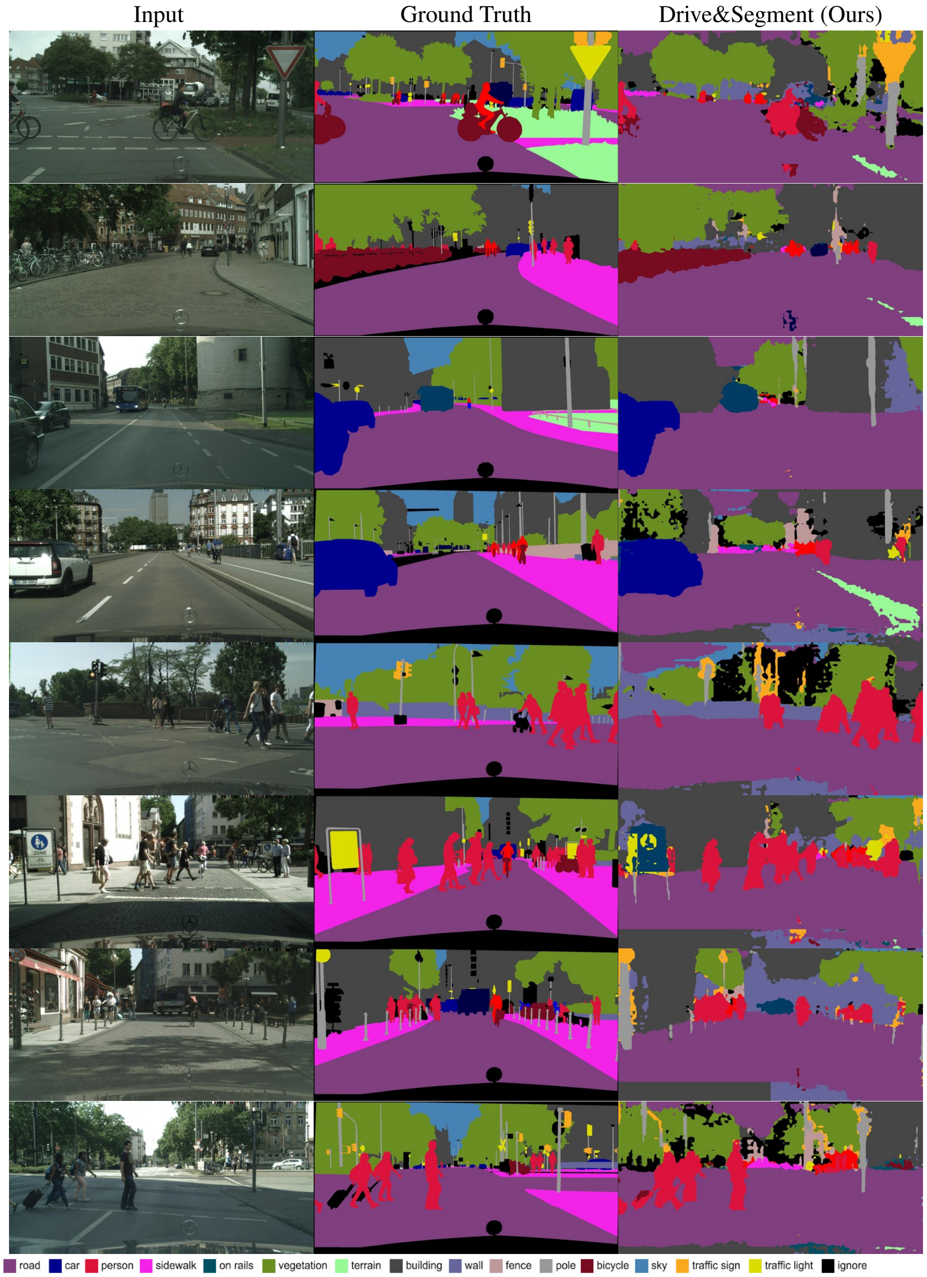


Fig. 13. Qualitative results for unsupervised semantic segmentation using our Drive&Segment approach on the validation split of the Cityscapes dataset. The matching between our pseudo-classes and the set of ground-truth classes is obtained using the Hungarian algorithm.

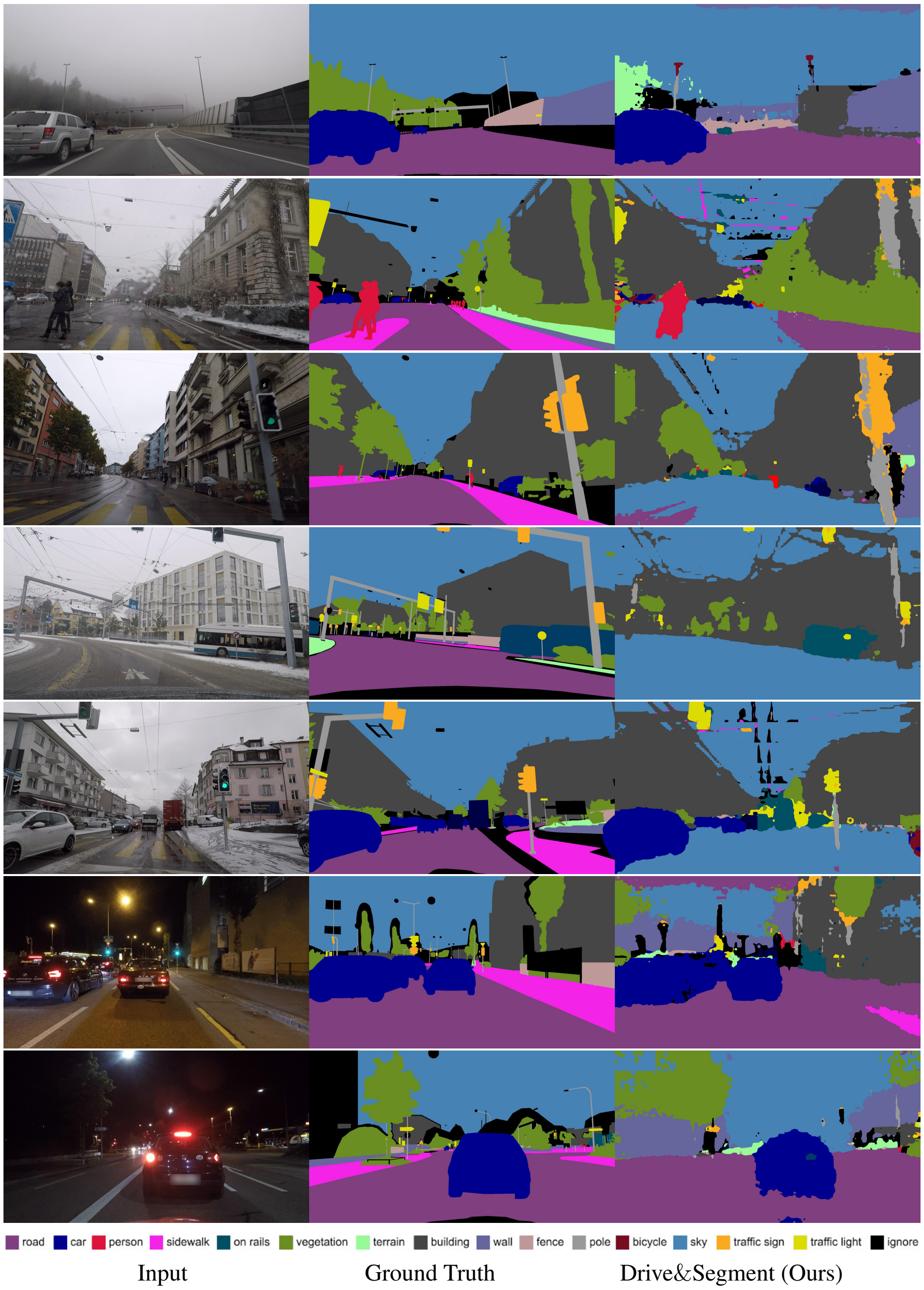


Fig. 14. Waymo Open Dataset *day* → ACDC [43] {*fog, rain, snow, night*}. Qualitative results of our Drive&Segment model trained on the daytime images from the Waymo Open Dataset and used to segment samples from the ACDC [43] dataset with various adverse conditions. In rows 2-5 the *ground* is incorrectly segmented as *sky*. This failure mode is further discussed in Section B.4.