Instance Segmentation with Cross-Modal Consistency

Alex Zihao Zhu[1], Vincent Casser[1], Reza Mahjourian[1], Henrik Kretzschmar[1] and Sören Pirk[2]

***Abstract*— Segmenting object instances is a key task in machine perception, with safety-critical applications in robotics and autonomous driving. We introduce a novel approach to instance segmentation that jointly leverages measurements from multiple sensor modalities, such as cameras and LiDAR. Our method learns to predict embeddings for each pixel or point that give rise to a dense segmentation of the scene. Specifically, our technique applies contrastive learning to points in the scene both across sensor modalities and the temporal domain. We demonstrate that this formulation encourages the models to learn embeddings that are invariant to viewpoint variations and consistent across sensor modalities. We further demonstrate that the embeddings are stable over time as objects move around the scene. This not only provides stable instance masks, but can also provide valuable signals to downstream tasks, such as object tracking. We evaluate our method on the Cityscapes and KITTI-360 datasets. We further conduct a number of ablation studies, demonstrating benefits when applying additional inputs for the contrastive loss.**

I. Introduction

Identifying and segmenting objects is a fundamental and challenging task in computer vision, and critically important for many robotics applications, such as autonomous driving. Given some sensor input such as a camera image or LiDAR scan, the goal of instance segmentation is to associate each pixel or point with a unique instance label for the object to which it corresponds. This allows for fine-grained reasoning about each object beyond sparse representations such as bounding boxes. Only recently, learning-based approaches have shown remarkable success in solving this task on a number of benchmark datasets [16], [6], [34]. Reliably identifying object instances in complex real-world data, both across different data modalities and coherent in time, remains a challenging and open problem [49], [15], [8].

Many safety-critical robotics systems, such as autonomous vehicles, rely on multiple sensors that provide complementary information to perceive the environment. For instance, LiDAR sensors provide highly accurate range readings, but only in clear weather conditions and up to a limited range. On the other hand, cameras may be able to perceive objects at much longer ranges, but without any explicit range readings and only in adequate lighting conditions. As a result, there is an opportunity to improve segmentation performance for each sensor by reasoning jointly across all modalities.

Another avenue is the utilization of temporal data, which can provide motion information, varying viewpoints and context for occlusions. Temporal data comes naturally for most sensors, although sequential labeling for ground truth is more expensive.

[1]Waymo LLC
[2]Adobe Research (Work done while at Google Research)

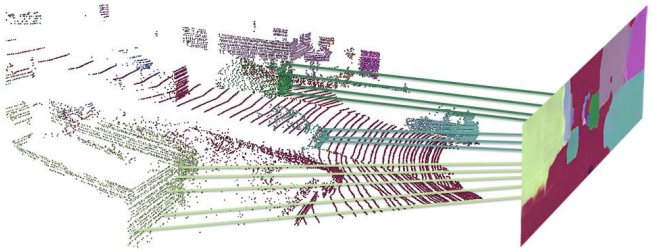

Fig. 1: Our approach to instance segmentation leverages information from multiple sensor modalities, such as 3D LiDAR point clouds and camera images (left). Our method learns to predict dense embeddings (depicted by the colors, right image) based on a novel cross-modal contrastive loss that uses samples from different sensor modalities and temporally consecutive frames of RGB images and LiDAR points. The resulting embeddings are stable over time and invariant to viewpoint changes.

In this paper, we aim to leverage these complementary signals and advance instance segmentation by training jointly on different data modalities and over time (Figure 1). Specifically, we propose a novel cross-modal contrastive loss that enables us to train with sequential samples from camera images and LiDAR point clouds to learn a dense representation of object instances. Our goal is to learn embeddings that coherently represent individual objects in both modalities. By training on different modalities, our method can more meaningfully disentangle individual instances. Moreover, we show that our loss formulation is flexible enough to be applied to sequences of sensor data of both modalities, enabling learning temporally-stable embeddings for object instances. For situations where only sequential data (but no labels) are available, we propose a method for generating pseudo-groundtruth using optical flow, and demonstrate that training on this data brings similar improvements in segmentation.

In summary, our contributions are as follows:

- We introduce a novel cross-modal contrastive loss to obtain consistent instance embeddings for RGB images and point clouds.
- We show that our loss can also be used to train on *sequences* of sensor data from both modalities, enabling the learning of temporally stable embeddings.
- We propose a method using optical flow to generate pseudo ground-truth for video, allowing the extension of our contrastive loss to partially labeled datasets.
- Our extensive experiments on the KITTI-360 [33] and Cityscapes [16] benchmarks suggest that our method significantly improves instance segmentation quality by up to 1.9 mAP when contrasting between temporal RGB data, and by up to 1.7 mAP when contrasting between temporal LiDAR and RGB data, while also improving the temporal stability of the embeddings.

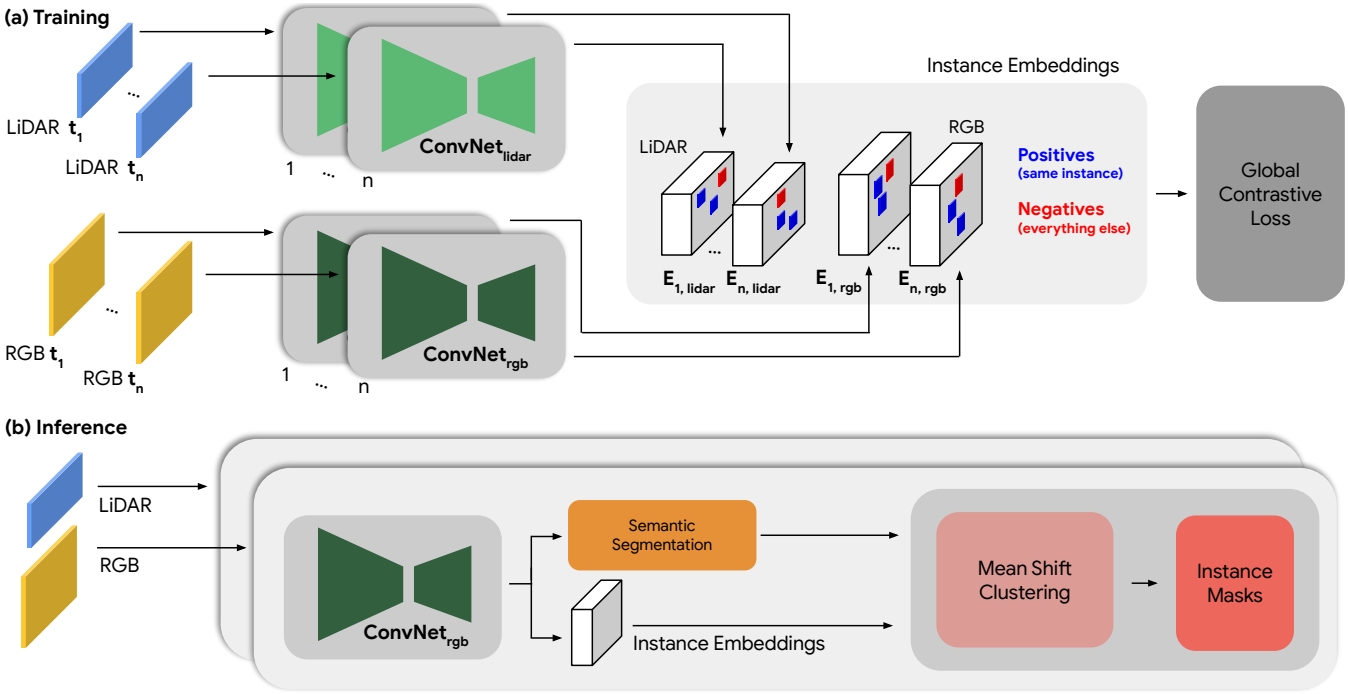

Fig. 2: Our goal is to learn dense cross-modal embeddings via contrastive learning to segment individual object instances. Our method leverages shared information between sensors and consecutive frames by applying the contrastive loss jointly over embeddings from all input sources at training time. This global contrastive loss allows the network to share information between different data sources, and improves overall segmentation performance. Once trained, the learned embeddings can be used to obtain object IDs through mean shift clustering.

II. Related Work

Object detection and segmentation have a long tradition in computer vision. While this spans a wide range of different approaches that we cannot conclusively discuss, we aim to provide an overview of learning-based methods toward instance and semantic segmentation in images and point clouds.

Semantic Segmentation. The goal of semantic segmentation is to associate every pixel in an image with a discrete label that defines its semantic class. A major challenge of effectively training networks for semantic segmentation is to define precise labels, and a number of different datasets were introduced for benchmarking [16], [22], [6]. Many recent methods for semantic segmentation employ deep neural network architectures for feature aggregation based on pyramid pooling [53], attention modules [10], multi-scale activations [24], or end-to-end convolutions [36]. Furthermore, it has been recognized that dilated or atrous convolutions provide a powerful way of capturing more global features through wider receptive fields [52], [9]. Other approaches aim to solve semantic segmentation with an emphasis on learning structured representations based on conditional random fields [1], [9] or deep parsing networks [35].

Instance Segmentation expands on the concept of semantic segmentation. Here the goal is to not only obtain a class label for every pixel, but to predict labels for individual object instances. Instance segmentation approaches can be separated into two different categories. In the first category, most methods rely on a two stage (or top-down) process [17], [18], [25]. A region proposal step [23] identifies object instances in a first stage and each region is then segmented into foreground and background during a second stage. The performance of top-down approaches is determined by obtaining region proposals and by the number of objects present in an image. On the contrary, bottom-up approaches aim to learn per-pixel embeddings that represent individual object instances based on recurrent instance grouping [29], watershed transformations [3], subnetwork instantiations [2], associative embeddings [39], or based on more refined loss functions such as discriminative loss [5], [26], or metric learning [20]. Bottom-up approaches are commonly implemented with more lightweight architectures, which offers more efficient training and easier integration into existing multi-task setups. However, learning meaningful embeddings from complex data to represent individual objects remains a challenging problem. Finally, a number of hybrid approaches exist that aim to learn embeddings of larger regions in images [7], [13].

3D Object Detection and Segmentation: Detecting and segmenting objects in point clouds has received a considerable amount of attention. PointNets [41] directly consume raw point clouds to generate latent representations that can be used for classification or semantic segmentation tasks. By leveraging various geometric relationships and representations a number of methods have since then improved upon the performance to solve for tasks, such as object detection [54], semantic segmentation [30], [42], or instance segmentation [45], [51]. A number of approaches rely on Bird's Eye View (BEV) representations to generate bounding boxes [50], [14] that – along with pillar-based representations [31], [46] – are popular in autonomous driving. More recent methods for 3D instance segmentation rely on more principled approaches. To this end 3D-MPA [19] uses an aggregation-based approach

where each point votes for its object center, HAIS [11] exploits the spatial relation of points and point sets, and SSTNet [32] uses a hierarchical representation which is constructed based on learned semantic features.

Multimodal Approaches. There also exist a limited number of multimodal approaches proposed in the robotic-vision domain, including [48], [37]. Most of these works tackle indoor environments using closely paired RGB-D data.

Similar to these methods, our goal is to leverage multi-modal sensor data. Specifically, we use a single-stage fully-convolutional network to learn per-point and per-pixel embeddings for instance segmentation. We train these embeddings with a metric learning loss that contrasts embeddings of the same object instance with those of different objects. The key idea to our approach is that we obtain the positive and negatives samples spatially, within the same frame, across different modalities (images, points), and even across time, in the adjacent frames of a sequence. We accomplish this by proposing a novel cross-modal consistency loss that allows us to sample embedding vectors from different sources. Training a single-stage network with this loss enables us to compute dense embeddings and, consequently, more robust instance masks for each frame, while also obtaining temporally stable results for sequences.

III. Method

In this section, we introduce our model architecture, the cross-modal contrastive loss function, and implementation details. Our full pipeline is shown in Figure 2.

A. Cross-Modal Contrastive Loss

Our method consists of three main components: (1) a pair of instance segmentation networks that generate dense embeddings for RGB images and LiDAR range images. (2) a sampling based contrastive learning loss which constrains embeddings from the same instances to be similar, while forcing embeddings from different instances to be different. (3) a consistency module based on paired instance labels that introduces additional training data and correspondences between sensors and over temporally neighboring frames.

Contrastive Loss: The goal of the instance segmentation network is to segment object instances by learning dense embedding vectors. The network takes as input either RGB images or LiDAR range images [4], and outputs embeddings with $c = 32$ channels. To train the network, we use a metric learning loss for contrasting embeddings of the same instance with those of different instances. Embeddings of the same instance are pushed closer together during training, while those of different instances are pushed away in the embedding space.

Our instance segmentation loss consists of the normalized temperature-scaled cross entropy (NT-XENT) loss defined in [12]. We define the per-pixel embeddings as $E_{h\times w\times c}$, where h and w denote the spatial dimensions of the output, and c, the channels of the embedding at each pixel. For each frame, we randomly sample $K_1 = 8192$ embeddings, distributed evenly across all instances (visualized for RGB

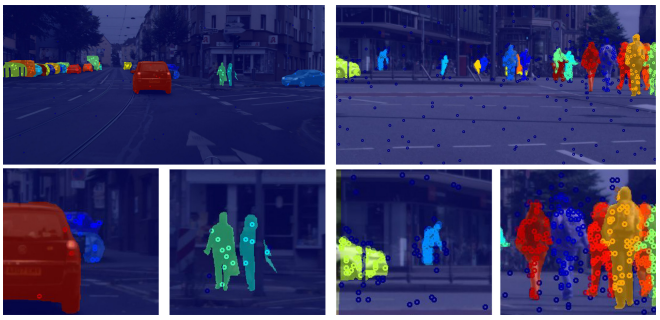

Fig. 3: Example of our sampling strategy where points are evenly distributed amongst instances. The bottom row shows close-ups of the used sample locations.

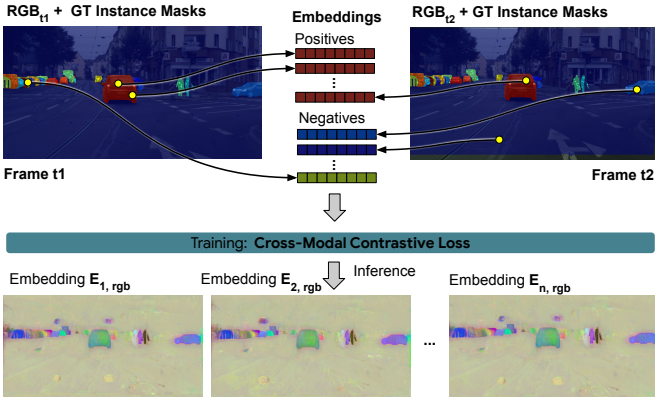

Fig. 4: Given a set of predicted embeddings with paired instance labels, we sample positive embedding vectors within the instance mask of an object in the current frame as well in all other frames (cross sensor and temporal). Samples from all other objects in all frames are used as negatives. We then train with our cross-modal contrastive loss to learn dense embeddings of object instances. The images at the bottom show the projected embeddings during training.

frames in Figure 3). Given a set of sampled embeddings for an object instance, we generate all possible positive pairs. The samples of all other instances are used as negatives for each pair. For a set of samples U and a positive pair, e_i, e_j, our loss can then be defined as:

$$l_{i,j} = -\log \frac{\exp(\text{sim}(e_i, e_j)/\tau)}{\sum_{k\in U} \mathbb{1}_{[id(i)\neq id(k))]} \exp(\text{sim}(e_i, e_k)/\tau)} \quad (1)$$

$$\text{with } \text{sim}(x, y) = \frac{x^T y}{\|x\|_2 \|y\|_2}, \quad (2)$$

where τ denotes a temperature hyperparameter and $id(i)$ is the mask ID of point i. The similarity function sim is used to compute the cosine similarity between two normalized embedding vectors x and y. The total loss over the instance embeddings is then defined as:

$$l_{\text{c}} = \frac{1}{\|U\|} \sum_{i,j\in U} \mathbb{1}_{[id(i)=id(j)]} l_{i,j}. \quad (3)$$

Furthermore, to obtain more stable results, we also apply a regularization loss over the norm of the embeddings:

$$l_{\text{r}} = \frac{1}{\|h \times w\|} \sum_{x,y} \|E(x, y)\|_2. \quad (4)$$

Our final loss, then, is:

$$l = l_c + \lambda l_r, \tag{5}$$

where $\lambda = 0.01$ is a weighting factor. Together, these losses allow us to learn meaningful embedding vectors as representation for object instances.

Consistency Over Modalities and Time: Given a dataset with instance labels which are consistent between sensors and across time, such that an object has the same ID across all inputs, one can apply the above training loss (5) between all inputs by randomly sampling points from each set of embeddings, and using the consistent instance labels to determine positive and negative pairs. This allows the network to learn embeddings which are consistent not only within a single frame, but between all modalities and time steps seen at training.

However, it is often the case that we do not have consistent instance labels. In these cases, we propose a method for generating pseudo-groundtruth using optical flow. Given a dataset with individually labeled frames amongst unlabeled temporal sequences, we use a pre-trained optical flow network to warp the groundtruth of the labeled frame to its temporal neighbors.

We use the predicted flow to warp the groundtruth instance and semantic labels of the current frame to the next frame and apply nearest neighbor resampling to avoid interpolating between integer label values. Following the work of Wang et al. [47], we compute occlusions according to the 'range map', defined by the number of points in the original image map to the other image. The instance and semantic labels of occluded pixels, as well as pixels that have left the image, are set to invalid, and are not sampled for the contrastive loss. The embeddings and warped labels for the next frame are then combined with those of the current frame when performing the sampling for the contrastive loss. Figure 4 illustrates the result of the warped labels as well as the sampling of embeddings from the two frames.

This pipeline has similarities to semi-supervised methods such as in the work by Chen et al. [8], which generate pseudo-groundtruth by running inference with a pre-trained model, and subsequently re-training. However, in addition to providing ground truth to temporally neighboring frames with optical flow, our method also ensures consistency between frames and thus over time. Sample outputs for our method over time can be found in Figure 5.

Semantic Segmentation: We predict a semantic mask for each frame and filter out all 'stuff' classes to a single background class. During clustering, we assign semantic classes to each clustered instance with a 'majority vote' between semantic predictions within the instance. We also generate the confidence for each mask using the average semantic score. Pixels not belonging to the selected semantic class are dropped from the instance to preserve semantic boundaries. The one-hot semantic labels are concatenated to the predicted instance embeddings, so that the network only has to predict unique instance embeddings within each class.

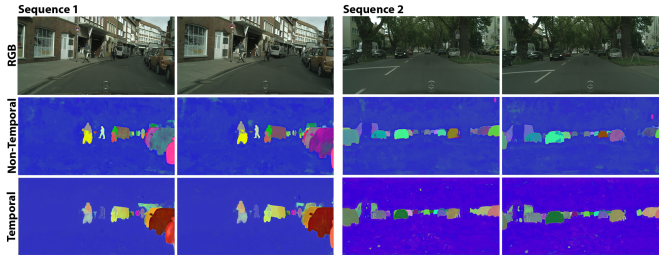

Fig. 5: Temporally-coherent embeddings: as our model is trained with a temporal consistency loss based on optical flow, we are able to obtain temporally-coherent and more stable instance embeddings. Here we show the RGB sequences of two different scenes along with the multi-dimensional per-pixel embedding vectors projected into a color space.

Instance Mask Generation:

At inference time, we cluster the resulting per-pixel embeddings by applying a variant of the mean shift algorithm proposed by Brabandere et al. [5]. In particular, we randomly sample a point in the embedding space, and find all inlier points with cosine distance (scaled to be within $[0, 1]$) less than a threshold, $m = 0.1$, from the sampled point. We then iterate by shifting the sampled point to the mean of the set of inliers, and repeat until convergence or some maximum number of iterations is reached. We repeat this process until all pixels have been clustered. With this method, we found that erroneous masks were typically generated on the transitions between masks, generating thin artifacts along the boundaries. To compensate, we filter out masks with an area to perimeter ratio less than a threshold, $r = 4$. As this method computes distances across the entire image, no assumptions about the connectivity of each mask are made, and so arbitrarily distributed instances can be detected.

More complex methods are available for generating instance masks from embeddings, such as the graph cut algorithm proposed in SSAP [21] or the transformer head proposed in MaX-DeepLab [44], which would likely achieve higher performance. However, these would incur additional latency penalties, and their computational power may hide improvements to the underlying embeddings.

B. Implementation Details

In this section, we provide details about the implementation of our network architecture.

Model Architecture: We use a modified architecture based on Panoptic DeepLab [15]. Our model consists of a Xception-71 backbone, with an additional ASPP and decoder module for instance embedding prediction, with $c = 32$ channels, in place of the instance center and offset decoder in [15]. For our cross-modal experiments, we use two separate networks for LiDAR range image inputs and RGB image inputs.

Preprocessing: For all datasets, we use full resolution images as input. During training, we augment our data by randomly flipping, scaling and cropping or padding the input images. Furthermore, we use random Gaussian blur and jitter brightness, contrast, saturation, and hue. For temporal data, we apply the same augmentation to each pair of images. For LiDAR range images, we only apply flipping, and ensure that the flipping is consistent with the camera images.

Method	All	Bicycle	Building	Car	Motorcycle	Person	Rider	Truck
Single camera	11.2	0.4	22.9	42.5	1.3	5.8	0.8	4.8
Temporal camera	11.9	1.4	**23.6**	**43.1**	2.0	6.5	0.8	6.1
Single camera + LiDAR	11.9	0.4	22.5	41.8	3.1	**6.9**	1.4	7.1
Temporal camera + LiDAR	12.2	**2.1**	22.7	41.0	3.0	6.4	2.9	7.2
Temporal camera + Temporal LiDAR	**12.9**	2.0	22.7	40.7	**5.4**	6.3	**4.4**	**8.7**

TABLE I: Ablation study on the effects of additional consistency on instance segmentation performance on a custom KITTI 360 train/val split. Adding additional inputs for the contrastive loss improves overall segmentation performance. Performance is measured in terms of AP (%).

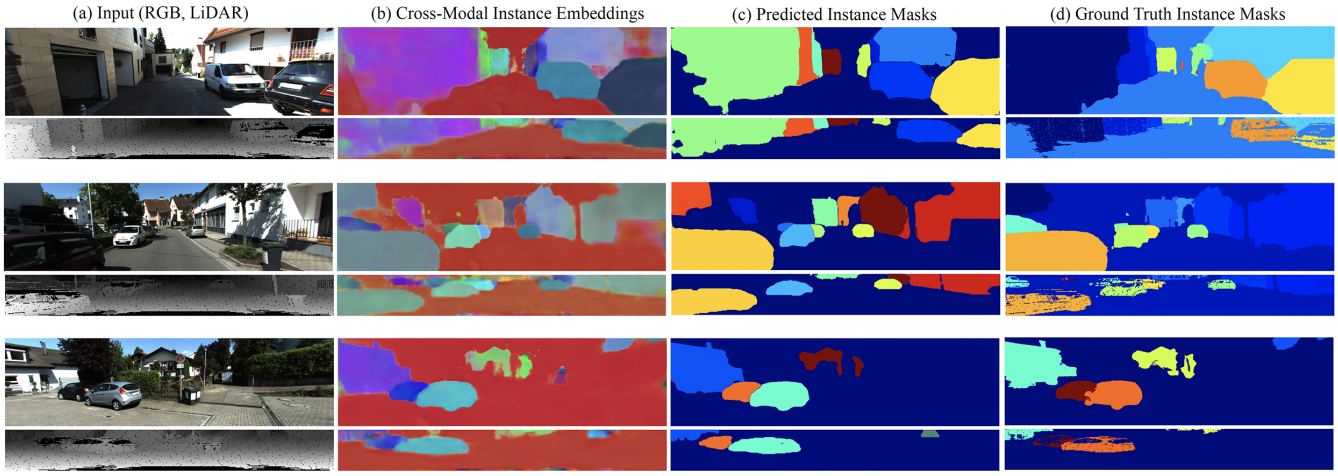

Fig. 6: Cross-modal results on KITTI-360: Our models take as inputs RGB images or LiDAR range images (a), and predicts dense instance embeddings (b) which are consistent between the sensor modalities. At post-processing, we can apply a mean-shift clustering algorithm to generate instance masks for each modality (c), which we compare to the ground truth (d). Our cross-modal contrastive loss contrasts pairs between the sensor modalities, and trains the network to predict the same embedding for each object instance in each sensor, denoted by similarity in color. Each image pair shows both sensor modalities, where the LiDAR data is shown as range images (bottom).

Training: For all experiments, we use a moving average ADAM optimizer with linear warmup for 5 epochs, initial learning rate of 2.5e-4 and exponential decay with decay factor of 1.2 every 10 epochs. For the KITTI-360 dataset, we train for 10,000 steps for the ablation train/val split, and 20,000 steps for the train/test split, with an effective batch size of 128. For the Cityscapes dataset, we train for 60,000 iterations with an effective batch size of 32.

IV. Experiments

In this section we present and discuss a variety of experiments, including baseline comparisons, ablation studies, and qualitative results our method is able to generate. For all of our experiments, we do not perform any kind of test time augmentation.

A. Datasets

a) KITTI-360: The KITTI-360 dataset [33] consists of camera and LiDAR panoptic segmentation and LiDAR bounding box labels for all of the sequences of the KITTI dataset [22]. For each sequence, the instance labels are consistent both between the camera and LiDAR labels, as well as between different time steps. The dataset contains 83,000 frames and associated LiDAR scans, split into 9 training sequences and 2 test sequences with held out groundtruth. As no validation set is provided, we perform our ablations on a train/val split where sequences 00, 02, 03, 04, 05, 06, 07 are the train set and sequences 09 and 10 are the validation set. The validation set was chosen from the two densest sequences, such that there are almost the same number of objects in the training and validation sets. For the final test set evaluation, we train on the entire training set.

As the dataset is largely transferred from primitive 3D shapes (cuboids, ellipsoids etc.), each label has a corresponding confidence value, which is used to weight the evaluation metrics. In accordance with the experiments in KITTI-360, we only use the top 70% of labels in each frame, in terms of confidence. To generate temporal frames, we randomly sample an image 2 frames before or after the current frame at each training step. To train a model on LiDAR, we use the range image representation of the point cloud generated by Bewley et al. [4]. From this 64×2048 range image, we crop a central 64×512 patch corresponding to the camera field of view, and treat this as an image into the DeepLab model. To generate per-scan groundtruth, we use nearest-neighbors to find the closest ground truth segmentation label to each point in the range image.

b) Cityscapes: To test our method on a dataset without consistent instance labels and in a single-modal setting, we employ the Cityscapes dataset [16]. dataset. Cityscapes provides us with 2975, 500, and 1525 images of urban scenes for training, testing and validation, respectively. Furthermore, the dataset provides 8 'thing' and 11 'stuff' classes. In this work, we train our network on the 'fine' set of training images with ground truth instance labels. In order to generate paired groundtruth, we use the unlabeled temporal frames,

and randomly sample frames from the images 2 frames before or after the labeled image. We then warp the groundtruth from the labeled frame to each temporal frame using a UFlow [27] model trained on the Waymo Open Dataset [43], consisting of 200,000 training images. To filter potential errors in the flow warping due to occlusions and other errors, we ignore warped labels based on the occlusion map generated by the flow model. For Cityscapes, we rely on a pre-trained semantic segmentation network for assigning class IDs, in particular an implementation of Panoptic Deeplab [15], with an Xception-71 backbone, trained on Cityscapes. Our implementation has a mIOU of 68.8% on the Cityscapes validation set. As embeddings along the boundary are particularly challenging, and the Cityscapes labels are typically reliable, we dedicate half of the points when sampling in the contrastive loss to embeddings that are within $b = 10$ pixels of the boundary of each instance mask.

B. Results on KITTI-360

In Table I, we report an ablation study on the effects of adding additional signals for the contrastive loss, both temporally and from different modalities. Overall, we observe that instance mAP for images improves as we add each additional frame to the consistency loss. The overall trend that we observe is that the network improves significantly for rare classes such as bicycle, motorcycle, rider and truck, while regressing slightly for the more common car class. This results in a significant overall mAP improvement. Note that, because the ablation models are trained on a much smaller training set, the mAP numbers in these experiments are naturally lower than those for the test set. We show qualitative examples from our best model (Temporal RGB + Temporal LiDAR) in Figure 6, where we demonstrate that the networks are able to predict embeddings that are consistent for objects across the sensor modalities. We also show our LiDAR instance segmentation results projected to 3D in Figure 7, demonstrating that the network can generate accurate 3D instance segmentation when operating only on the range image. However, we did not observe similar improvements in 3D LiDAR mAP when training with the proposed cross-modality contrastive loss. Our observation is that, as the 2D labels are generated by projecting the 3D labels (and applying a CRF), they are typically much noisier than the 3D labels, and cannot provide useful signal for the 3D part of the model.

In Table II, we report a comparison in the mAP of our final method against the popular MaskRCNN method [25] on the held-out test set, where we outperform the ResNet-50 baseline, while approaching the performance of the ResNet-101 baseline. At the time of writing, the validation set for this dataset was not available, and better results will be achievable once the same training splits are available as for the baselines.

Method	AP (%)
Mask-RCNN ResNet-50	19.5
Mask-RCNN ResNet-101	20.9
Ours with temporal + lidar temporal	20.3

TABLE II: Results on the KITTI-360 held out test set.

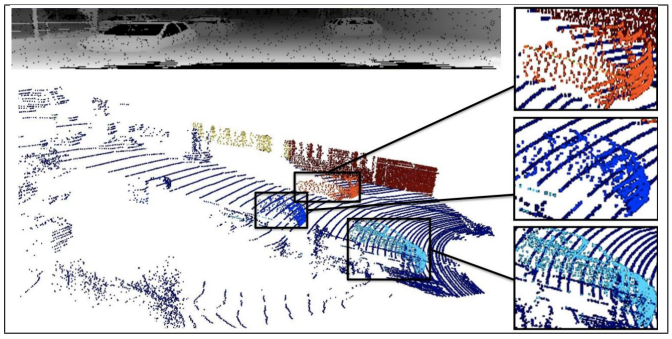

Fig. 7: Close-up analysis of our 3D LiDAR segmentation network, operating on range images (top-left), demonstrating accurate point-cloud instance segmentation.

Method	Approach	Input Size	AP (%)	Speed (ms)
Panoptic FPN [28]	TD	512× 1024	33.2	-
UPSNet [49]	TD	1024×2048	33.3	202ms
Seamless [40]	TD	1024× 2048	33.6	-
Panoptic Deeplab [15] Xception-71	TD[1]	1025×2049	35.3	175ms
SSAP [21]	BU	1024× 2048	31.5	260ms+[2]
Ours base	BU-C	1024× 2048	29.0	182ms
Ours temporal	BU-C	1024× 2048	30.9	182ms

TABLE III: Cityscapes validation set results, without any test time augmentation or additional training data for all methods. TD: Top-Down. BU: Bottom-Up. BU-C: Bottom-Up via Contrastive Learning. [1]Panoptic Deeplab is a single stage network, but relies on top-down object proposals of object centers. [2]Reported time is for post-processing only and does not include the network forward pass.

C. Results on Cityscapes

In Table III, we report our instance segmentation results on Cityscapes compared to existing methods. The reported results were obtained on the Cityscapes validation set. To provide a fair comparison of the main methods themselves, we use the reported numbers for each method without test time augmentation. In addition, we report mAP for our method using groundtruth semantic segmentations in Table IV, and compare against past work which has used a similar metric. This approach allows us to evaluate the instance segmentation network independently from the performance of the semantic segmentation. Overall, the improvement from our cross-modal consistency loss allows us to strongly outperform bottom up methods such as Brabandere et al. [5] and Neven et al. [38] with groundtruth semantics. It also pushes our embedding-based method close to the previous bottom up works such as SSAP [21]. However, this work requires a complex post-processing procedure which has a significant latency, as reported in Table III. We underperform compared to Panoptic DeepLab, but note that this method, while being single shot, requires the prediction of object centers. These centers serve as top-down proposals requiring instances to have distinguishable centers and some form of NMS.

We also show visualizations of our predictions in Figure 8, where the embeddings can robustly and precisely represent a large variety of object instances.

Temporal Constraints: To validate the impact of the temporal contrastive loss, we trained our model with and without contrasting over time (see Table III, Ours base vs Ours temporal). From these results, introducing the contrastive

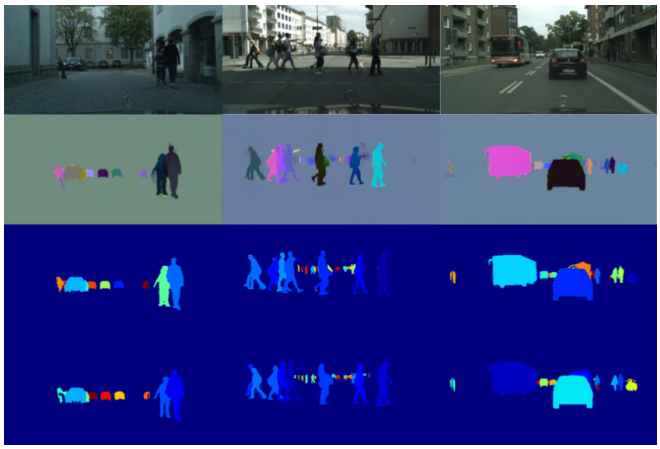

Fig. 8: Visualization of per-pixel embedding vectors on Cityscapes: our method allows us to obtain per-pixel embedding vectors from RGB input images (first row). The embeddings robustly and precisely disentangle object instances across different classes (third row). To visualize the embeddings, we project them into RGB color space (second row). In the last row, we show the ground truth instance masks. Please note that the color of the predicted and ground truth masks are not supposed to match.

Method	Approach	Input Size	AP (%)	Speed (ms)
Brabandere et al. [5] (ResNet-38)	BU-C	384× 768	29.0	200ms
Neven et al. [38]	BU	1024× 2048	40.5	91ms
Ours base	BU-C	1024× 2048	48.9	182ms
Ours temporal	BU-C	1024× 2048	50.6	182ms

TABLE IV: Cityscapes validation set results using groundtruth semantics for assigning classes to each instance. BU: Bottom-Up. BU-C: Bottom-Up via Contrastive Learning. For this setup, we significantly outperform other contrastive learning methods.

loss over time has a significant impact on the predicted instances masks. Adding temporal frames to the contrastive loss improves overall AP by 2%. The visualization of the predicted embeddings in Figure 5 also shows fewer artifacts.

We also compute error metrics on the stability of the instance embeddings over time for the Cityscapes dataset. Given two frames, we warp the predicted embeddings of the next frame to the current frame, and compute the cosine distance between the average embedding of each instance in the two time steps, while ignoring any pixels estimated as 'occlusion' from the optical flow. In addition, we compute as accuracy the proportion of instance pairs with cosine distance less than the clustering threshold $m = 0.1$. These results can be found in Table V. We observe that the majority of embeddings (75.1%) are stable over time, even without the proposed temporal contrastive loss. However, there are a number of cases, such as large motions and dense objects, where the embeddings are not stable. Overall, our proposed loss reduces the cosine distance between instances by 0.025, and improves accuracy by 8.3% to 83.4%.

Embedding Similarity: In Figure 9 we show the results of an embedding similarity experiment. We select pixels in the image and compute the distance, shown in grayscale, of the selected embeddings with all other embeddings in the image. As illustrated, the learned embeddings allow us to disentangle object instances of the same class, across different classes, and the background. Our simple clustering scheme can generate high quality masks by thresholding this similarity.

Method	Cosine Distance ⇓	Accuracy ⇑
Without Temporal Loss	0.087	75.1
With Temporal Loss	0.062	83.4

TABLE V: Temporal consistency metrics. We compute the cosine distance (in $[0, 1]$) between the average embedding for each instance between two temporal frames. Accuracy is the percentage of instances with cosine distance $< m = 0.1$ over time. Training with temporal consistency significantly increases performance.

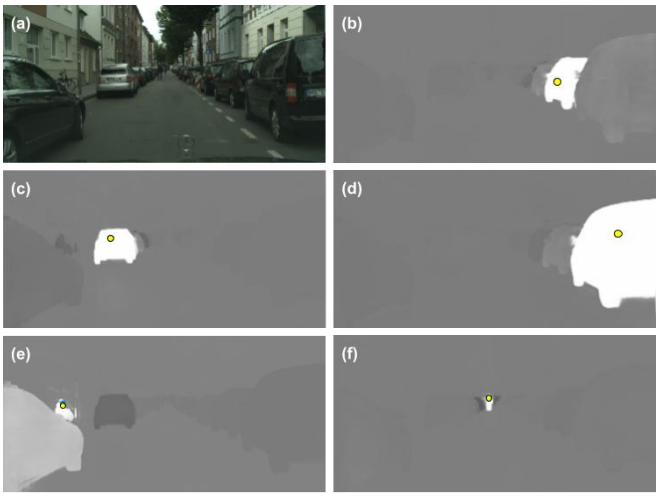

Fig. 9: Pixel similarity: for an image with objects of different classes (a) we randomly select different pixel positions in the image (yellow dot) and show the similarity of the selected embedding to all other embeddings as gray-scale values. As shown, the learned embeddings allow us to meaningfully disentangle object instances, such as various different cars (b, c, d), intricate objects like bicycles (e), and small-scale objects, such as pedestrians (f).

V. Limitations and Future Work

The main contribution of this method is the contrastive learning of instance embeddings between sensors and over time. As it stands, we believe that the learned embeddings are strong, but have chosen a relatively simple method to generate instance masks for evaluation. We use this post-processing to highlight the improvements in the embeddings themselves, where a more complex method may obscure such improvements. However, there is room for improvement if stronger AP scores are desired, by adding a graph cut optimization such as in SSAP [21] or a transformer head on top of the embeddings as in MaX-DeepLab [44].

VI. Conclusion

We have introduced a novel method for learning pixel-wise embeddings as a representation for individual object instances. Compared to other bottom-up approaches, we employ a contrastive learning scheme; embeddings of the same object instance are pushed closer together, while they are pulled away from other objects and the background. In particular, we leverage information between sensors, as well as over time, by applying our cross-modal contrastive loss on the union of all predicted embeddings for a given scene. Through quantitative experiments, we show that this contrastive loss is able to significantly improve instance segmentation performance, and coherent instance embeddings between sensors and time, which are clustered to generate high quality instance masks.

REFERENCES

[1] A. Arnab, S. Jayasumana, S. Zheng, and P. Torr. Higher order conditional random fields in deep neural networks. In *ECCV*, 2016.

[2] A. Arnab and P. Torr. Pixelwise instance segmentation with a dynamically instantiated network. *CVPR*, pages 879–888, 07 2017.

[3] M. Bai and R. Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, pages 2858–2866, 2017.

[4] A. Bewley, P. Sun, T. Mensink, D. Anguelov, and C. Sminchisescu. Range conditioned dilated convolutions for scale invariant 3d object detection. *arXiv*, 2020.

[5] B. De Brabandere, D. Neven, and L. Gool. Semantic instance segmentation with a discriminative loss function. *ArXiv*, 2017.

[6] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv*, 2019.

[7] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan. BlendMask: Top-down meets bottom-up for instance segmentation. In *CVPR*, 2020.

[8] L.-C. Chen, R. Gontijo Lopes, B. Cheng, M. D. Collins, E. D. Cubuk, B. Zoph, H. Adam, and J. Shlens. Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation. In *ECCV*, volume 12354, pages 695–714. Springer, 2020.

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. 2016.

[10] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. Yuille. Attention to scale: Scale-aware semantic image segmentation. *CVPR*, pages 3640–3649, 2016.

[11] S. Chen, J. Fang, Q. Zhang, W. Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation, 2021.

[12] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. *arXiv*, 2020.

[13] X. Chen, R. Girshick, K. He, and P. Dollár. Tensormask: A foundation for dense object segmentation, 03 2019.

[14] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. pages 6526–6534, 07 2017.

[15] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *CVPR*, 2020.

[16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[17] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance-sensitive fully convolutional networks. In *ECCV*, pages 534–549, 2016.

[18] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *CVPR*, pages 3992–4000, 2015.

[19] F. Engelmann, M. Bokeloh, A. Fathi, B. Leibe, and M. Nießner. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. *CVPR*, pages 9028–9037, 2020.

[20] A. Fathi, Z. W., V. Rathod, P. Wang, H. O. Song, S. Guadarrama, and K. Murphy. Semantic instance segmentation via deep metric learning. *ArXiv*, abs/1703.10277, 2017.

[21] N. Gao, Y. Shan, Y. Wang, X. Zhao, Y. Yu, M. Yang, and K. Huang. Ssap: Single-shot instance segmentation with affinity pyramid. *CoRR*, abs/1909.01616, 2019.

[22] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013.

[23] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[24] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015.

[25] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[26] A. Hu, A. Kendall, and R. Cipolla. Learning a spatio-temporal embedding for video instance segmentation. 2019.

[27] R. Jonschkowski, A. Stone, J. Barron, A. Gordon, K. Konolige, and A. Angelova. What matters in unsupervised optical flow. *ECCV*, 2020.

[28] A. Kirillov, R. Girshick, K. He, and P. Dollár. Panoptic feature pyramid networks. *arXiv preprint arXiv:1901.02446*, 2019.

[29] S. Kong and C. Fowlkes. Recurrent pixel embedding for instance grouping. In *CVPR*, 2018.

[30] L. Landrieu and M. Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, pages 4558–4567, 2018.

[31] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. Pointpillars: Fast encoders for object detection from point clouds. *CVPR*, pages 12689–12697, 2019.

[32] Z. Liang, Z. Li, S. Xu, M. Tan, and K. Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *CVPR*, pages 2783–2792, 2021.

[33] Y. Liao, J. Xie, and A. Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *arXiv*, 2021.

[34] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft coco: Common objects in context, 2014.

[35] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *ICCV*, page 1377–1385, USA, 2015. IEEE Computer Society.

[36] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[37] Johannes Meyer, Andreas Eitel, Thomas Brox, and Wolfram Burgard. Improving unimodal object recognition with multimodal contrastive learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5656–5663. IEEE, 2020.

[38] D. Neven, B. De Brabandere, M. Proesmans, and L. Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *CVPR*, pages 8837–8845, 2019.

[39] A. Newell, Z. Huang, and J. Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *NIPS*, page 2274–2284, 2017.

[40] L. Porzi, S. Rota Bulò, A. Colovic, and P. Kontschieder. Seamless scene segmentation. 2019.

[41] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. 2016.

[42] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, volume 30. Curran Associates, Inc., 2017.

[43] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, pages 2446–2454, 2020.

[44] H. Wang, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *CVPR*, pages 5463–5474, 2021.

[45] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia. Associatively segmenting instances and semantics in point clouds. In *CVPR*, pages 4091–4100, 2019.

[46] Y. Wang, A. Fathi, A. Kundu, D. A. Ross, C. Pantofaru, T. A. Funkhouser, and J. M. Solomon. Pillar-based object detection for autonomous driving. In *ECCV*, 2020.

[47] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, and W. Xu. Occlusion aware unsupervised learning of optical flow. In *CVPR*, pages 4884–4893, 2018.

[48] Yu Xiang, Christopher Xie, Arsalan Mousavian, and Dieter Fox. Learning rgb-d feature embeddings for unseen object instance segmentation. *CoRL*, 2020.

[49] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, and R. Urtasun. Upsnet: A unified panoptic segmentation network. In *CVPR*, 2019.

[50] B. Yang, W. Luo, and R. Urtasun. Pixor: Real-time 3d object detection from point clouds. *CVPR*, pages 7652–7660, 2018.

[51] L. Yi, W. Zhao, H. Wang, M. Sung, and L. J. Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. *CVPR*, pages 3942–3951, 2019.

[52] F. Yu, V. Koltun, and T. Funkhouser. Dilated residual networks. In *CVPR*, pages 636–644, 2017.

[53] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, pages 6230–6239, 2017.

[54] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. *CVPR*, pages 4490–4499, 2018.