CoDEPS: Online Continual Learning for Depth Estimation and Panoptic Segmentation

Niclas Vödisch[1*], Kürsat Petek[1*], Wolfram Burgard[2], and Abhinav Valada[1]
[1]University of Freiburg [2]University of Technology Nuremberg

***Abstract*—Operating a robot in the open world requires a high level of robustness with respect to previously unseen environments. Optimally, the robot is able to adapt by itself to new conditions without human supervision, e.g., automatically adjusting its perception system to changing lighting conditions. In this work, we address the task of continual learning for deep learning-based monocular depth estimation and panoptic segmentation in new environments in an online manner. We introduce CoDEPS to perform continual learning involving multiple real-world domains while mitigating catastrophic forgetting by leveraging experience replay. In particular, we propose a novel domain-mixing strategy to generate pseudo-labels to adapt panoptic segmentation. Furthermore, we explicitly address the limited storage capacity of robotic systems by proposing sampling strategies for constructing a fixed-size replay buffer based on rare semantic class sampling and image diversity. We perform extensive evaluations of CoDEPS on various real-world datasets demonstrating that it successfully adapts to unseen environments without sacrificing performance on previous domains while achieving state-of-the-art results. The code of our work is publicly available at http://codeps.cs.uni-freiburg.de.**

I. Introduction

Deploying robots such as autonomous cars in urban scenarios requires a holistic understanding of the environment with a unified perception of semantics, instances, and depth. The joint solution of these tasks enables vision-based methods to generate a 3D semantic reconstruction of the scene, which can be leveraged for downstream applications such as localization or planning. While deep learning-based state-of-the-art approaches perform well when inference is done under similar conditions as used for training, their performance can drastically decrease when the new target domain differs from the source domain, e.g., due to environmental conditions [31], different sensor parameters [3, 6]. This domain gap poses a great challenge for robotic platforms that are deployed in the open world without prior knowledge about the target domain. Additionally, unlike the source domain where ground truth annotations are generally assumed to be known and can be used for the initial training, such supervision is not applicable to the target domain due to the absence of labels, rendering classical domain adaptation methods unsuitable. Unsupervised domain adaptation attempts to overcome these limitations. However, the vast majority of proposed approaches focus on sim-to-real domain adaptation mostly in an offline manner [12, 23], i.e., a directed knowledge transfer without the need to avoid catastrophic forgetting and with access to abundant target annotations. Additionally, such works often do not consider limitations on a robotic platform, e.g., available compute hardware and limited storage capacity [17, 32].

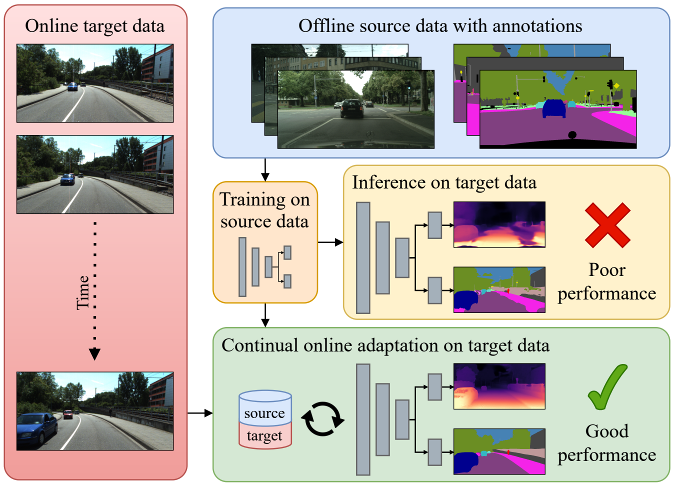

Fig. 1. Neural networks often perform poorly when deployed on a target domain that differs from the source domain used for training. To close this domain gap, we propose to continuously adapt the network by exploiting online target images. To mitigate catastrophic forgetting and enhance generalizability, we leverage a fixed-size replay buffer allowing the method to revisit data from both the source and target domains.

In this work, we use online continual learning to address these challenges for depth estimation and panoptic segmentation in a multi-task setup. In particular, we leverage images from an onboard camera to perform online continual learning enhancing performance during inference time. While a naive approach would result in overfitting to the current scene, our method CoDEPS mitigates forgetting by leveraging experience replay of both source data and previously seen target images. We combine a classical replay buffer with generative replay in the form of a novel cross-domain mixing strategy allowing us to exploit supervised ideas also for unlabeled target data. Unlike existing works, we explicitly address the aforementioned hardware limitations by using only a single GPU and restricting the replay buffer to a fixed size. We demonstrate that CoDEPS successfully improves on new target domains without sacrificing performance on previous domains.

The main contributions of this work are as follows:

1) We introduce CoDEPS, the first online continual learning approach for joint monocular depth estimation and panoptic segmentation.
2) We propose a novel cross-domain mixing strategy to adapt panoptic segmentation to unlabeled target data.

[*] Equal contribution.

3) To address the memory restrictions of robotic platforms, we develop a fixed-size replay buffer based on rare class sampling and image diversity.
4) We extensively evaluate CoDEPS and compare it to other methods in challenging real-to-real settings.
5) We release our code and the trained models at http://codeps.cs.uni-freiburg.de.

II. Related Work

In this section, we provide an overview of monocular depth estimation, panoptic segmentation, and unsupervised domain adaptation including continual learning.

Monocular Depth Estimation: Monocular depth estimation is the task of predicting a dense depth map from a single RGB image. While supervised approaches exploit measurements from range sensors to supervise the network predictions [29], unsupervised methods leverage geometric cues from temporal context [9, 36]. Most of the research on unsupervised learning tackles the limitations of the so-called photometric loss function that is usually employed for unsupervised depth learning, e.g., dynamic object handling [2, 4, 19], occlusion [10], and abrupt illumination changes [34]. In this work, we leverage Monodepth2 [10] for unsupervised depth learning and employ it similarly to Guizilini et al. [12] for the purpose of domain adaptation.

Panoptic Segmentation: Panoptic segmentation unifies the two tasks of semantic and instance segmentation by fusing the respective targets into a joint panoptic output. Furthermore, semantic classes are grouped into "stuff" classes, e.g., *road* or *building*, and "thing" classes, e.g., *car* or *pedestrian*. In particular, the goal of vision-based panoptic segmentation is to assign a semantic class and for "thing" classes an additional instance label to each pixel of an image. Panoptic segmentation networks usually comprise a joint encoder and separate decoders for each subtask, whose outputs are subsequently merged by a panoptic fusion module. Existing works can be categorized into bottom-up [5, 27] and top-down [11, 26] approaches. Whereas bottom-up methods detect instances in a proposal-free manner from the semantic prediction, top-down methods include an additional proposal generation step. Contradictions to the semantic predictions are then resolved during post-processing. In this work, we build upon the bottom-up Panoptic-Deeplab [5] with changes to the semantic head according to [12].

Unsupervised Domain Adaptation: Domain adaptation aims to bridge the domain gap between a source domain $\mathcal{S}$ used for training and a target domain $\mathcal{T}$ used for inference to mitigate a loss in performance. An important aspect is whether the performance on the source domain must be maintained, linking domain adaptation to continual learning (CL) [22], where the objective of a task or the task itself can change over time. A CL system has to adapt to the new target objective while retaining the knowledge to solve the previous task(s), i.e., avoiding catastrophic forgetting. Ideally, the CL system can further achieve positive forward transfer, i.e., improve on future yet untrained tasks given the current task. In many real-world scenarios ground truth annotations for the target domain are not available, thus requiring unsupervised domain adaptation (UDA) methodology. Offline UDA assumes that abundant target data is accessible. However, in order to guarantee the continuous operation of a robot in new domains, UDA approaches have to work online without previous target data collection.

Offline UDA can leverage both annotated source data and abundant unlabeled target data, enabling learning a given task from $\mathcal{S}$ while simultaneously adapting the network to $\mathcal{T}$. For depth estimation, DESC [23] adapts from a synthetic source domain containing RGB images and ground truth depth to a real-world target domain by performing source-to-target style transfer and using a consistency loss between depth predictions from RGB and semantic maps. GUDA [12] tackles UDA for semantic segmentation using depth estimation as a proxy task. A shared encoder with task-specific heads for depth estimation and semantic segmentation is trained via source supervision. Simultaneously, data from $\mathcal{T}$ is used to update the encoder and depth head in an unsupervised manner. Due to the refined weights of the encoder, the semantic predictions on $\mathcal{T}$ improve as well. Another common approach for adapting semantic segmentation is cross-domain sampling enabling partial supervision on $\mathcal{T}$. DACS [30] mixes images from $\mathcal{S}$ and $\mathcal{T}$ by copying the pixels of a source image to a target image based on the semantic labels [28]. The semantic prediction of the target image is updated with ground truth source labels for the same set of pixels. The network is then jointly trained on annotated source data and the pseudo-labeled mixing data. Recently, ConfMix [24] proposed a simple yet effective mixing strategy for object detection, where a target image is divided into rectangular image regions. The region with the most confident predictions is then copied onto a source image and the respective ground truth annotations. Finally, Huang et al. [15] propose a UDA method for panoptic segmentation by regularizing complementary features from semantic and instance segmentation. In this work, we extend the aforementioned mixing strategies to instance-based sampling and explicitly address differing camera parameters.

During online UDA, samples from $\mathcal{T}$ can only be accessed in a consecutive manner resembling the image stream of a camera. Typically, a network is trained offline via supervision on $\mathcal{S}$ and then adapted to $\mathcal{T}$ during inference time. Such a setup rises two main challenges: first, incoming target samples originate from highly similar scenes and thus drastically reduce the diversity; second, this similarity of consecutive samples leads to a strong overfitting of the model to the scene [35]. Initial works for online UDA focused on depth learning [17, 35] and visual odometry [20, 32], for which unsupervised training schemes are already well established. Whereas Zhang et al. [35] propose novel network modules that are adapted via a meta-learning paradigm to mitigate forgetting, CoMoDA [17] employs a common CL strategy, i.e., experience replay to combine the online target sample with previously seen samples. Continual SLAM [32] also uses unsupervised depth estimation as a proxy task to enhance visual odometry during inference time. Additionally, it demonstrates that incorporating samples from $\mathcal{S}$ and previous target domains $\mathcal{T}_i$ prevents catastrophic forgetting when revisiting domains. Similar settings involving multiple target domains, which are hence closely related to classical CL, are also addressed for semantic segmentation. CBNA [16] mixes statistics from $\mathcal{S}$ and $\mathcal{T}$ to update the batch normalization layers and showcases the efficacy of the approach on continually visited target domains. CoTTA [33] adapts the entire network without using source data but self-supervision. To tackle error accumulation, it uses an exponential moving average filter and student-teacher consistency when updating the network weights. Using depth estimation as a proxy task, Kuznietsov et al. [18] extend GUDA [12] to online UDA with experience replay and confidence regularization on the semantic predictions. To the best of our knowledge, we propose the first approach for online continual UDA for joint depth estimation and panoptic segmentation.

III. Technical Approach

The setting investigated in this work consists of two steps. First, we train a neural network on the source domain $\mathcal{S}$ partly using ground truth supervision. Second, to close the gap between domains, we continuously adapt the network during inference time on the target domain $\mathcal{T}$ using a replay buffer and unsupervised training strategies.

A. Network Architecture and Source Domain Pretraining

In this section, we detail the network architecture and loss functions that we employ during the pretraining stage on the source domain.

Architecture: We build our network following a common multi-task design scheme, i.e., using a single backbone followed by task-specific heads. A high-level overview of the network architecture is shown in Fig. 2. In detail, we use a ResNet-101 [13] as the shared encoder for all three tasks including depth prediction, semantic segmentation, and instance segmentation. The depth head follows the design of Monodepth2 [10] comprising five consecutive convolutional layers with skip connections to the encoder. Additionally, we include a separate PoseNet consisting of a ResNet-18 encoder and a four-layer CNN to estimate the camera motion between two image frames. For panoptic segmentation, we follow the bottom-up method Panoptic-Deeplab [5], leveraging separate heads for semantic segmentation and instance segmentation, and slightly modify the semantic head [12]. Specifically, the instance head consists of two sub-heads to predict the center of an object and to associate each pixel of an image to the corresponding object or the background. Combining both the semantic and instance predictions, a panoptic fusion module generates the overall panoptic segmentation map.

Source Domain Pretraining: During the initial training phase on the source domain, we assume to have access to image sequences as well as ground truth panoptic segmentation annotations. In the following, we briefly describe the respective loss functions that we employ for training the three task-specific heads.

We train the depth estimation head using the common methodology of unsupervised training based on the photometric error [10]. In particular, we leverage an image triplet $\{\mathbf{I_{t_0}}, \mathbf{I_{t_1}}, \mathbf{I_{t_2}}\}$ to predict depth $\mathbf{D_{t_1}}$ and camera motion $\mathbf{M_{t_0 \to t_1}}$ and $\mathbf{M_{t_1 \to t_2}}$. Afterwards, we compute the photometric error loss $\mathcal{L}^d_{pe}$ as a weighted sum of the reprojection loss $\mathcal{L}^d_{pr}$ and the image smoothness loss $\mathcal{L}^d_{sm}$:

$$\mathcal{L}^d_{pe} = \lambda_{pr}\mathcal{L}^d_{pr} + \lambda_{sm}\mathcal{L}^d_{sm}. \tag{1}$$

We train the semantic segmentation head in a supervised manner using the bootstrapped cross-entropy loss with hard pixel mining $\mathcal{L}^{sem}_{bce}$ following Panoptic-Deeplab [5].

For training the instance segmentation head, we adopt the MSE loss $\mathcal{L}^{ins}_{center}$ for the center head and the L1 loss $\mathcal{L}^{ins}_{offset}$ for the offset head. The total loss to supervise instance segmentation is then computed as a weighted sum:

$$\mathcal{L}^{ins}_{co} = \lambda_{center}\mathcal{L}^{ins}_{center} + \lambda_{offset}\mathcal{L}^{ins}_{offset}. \tag{2}$$

B. Online Adaptation

After the described network has been trained on the source domain $\mathcal{S}$ using the aforementioned losses, we aim to adapt it to the target domain $\mathcal{T}$ in a continuous manner. That is, unlike other works, data from the target domain is revealed frame by frame resembling the online stream of an onboard camera. As depicted in Fig. 2, every adaptation iteration consists of the following steps:

1) Construct an update batch by combining online and replay data.
2) Generate pseudo-labels using the proposed cross-domain mixing strategy.
3) Perform backpropagation to update the network weights.
4) Update the replay buffer.

In this section, we first detail the structure of the utilized replay buffer and then propose adaptation schemes for both depth estimation and panoptic segmentation.

Replay Buffer and Batch Generation: Upon receiving a new image taken by the robot's onboard camera, we construct a batch that is used to perform backpropagation on the network weights. In detail, a batch $\mathbf{b_t}$ consists of the current online image $\mathbf{I_t} \in \mathcal{T}$, previously received target images $\mathbf{I}_{\mathcal{T}_\mathbf{i}} \in \mathbf{B}_\mathcal{T}$, and fully annotated source samples $\mathbf{I}_{\mathcal{S}_\mathbf{i}} \in \mathbf{B}_\mathcal{S}$. Here, $\mathbf{B}_\mathcal{T} \subseteq \mathcal{T}$ and $\mathbf{B}_\mathcal{S} \subseteq \mathcal{S}$ denote the respective replay buffers. Formally[*],

$$\mathbf{b_t} = \{\mathbf{I_t}, \mathbf{I}_{\mathcal{T}_0}, \mathbf{I}_{\mathcal{T}_1}, \dots, \mathbf{I}_{\mathcal{S}_0}, \mathbf{I}_{\mathcal{S}_1}, \dots\}. \tag{3}$$

By revisiting target images from the past, we increase the diversity in the loss signal on the target domain and hence mitigate overfitting to the current scene. This further accounts for situations in which the current online image suffers from visual

[*] To improve readability, we omit in the notation that each image sample includes its two previous frames enabling unsupervised depth estimation.

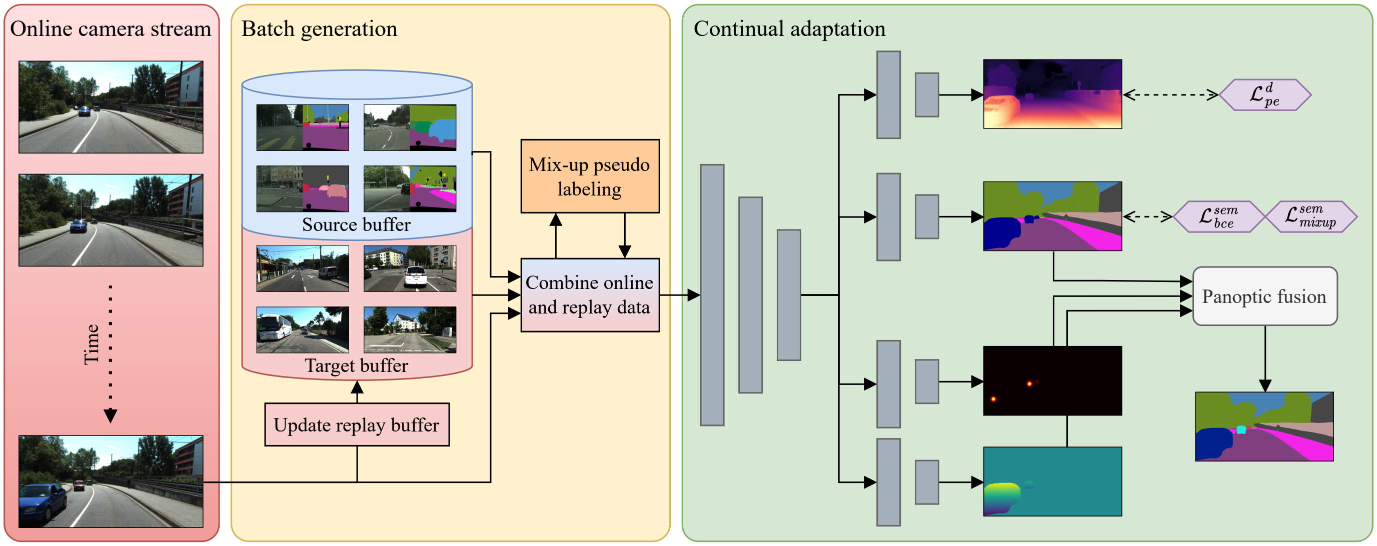

Fig. 2. Overview of our proposed CoDEPS. Unlabeled RGB images from an online camera stream are combined with samples from a replay buffer comprising both annotated source samples and previously seen target images. Cross-domain mixing enables pseudo-supervision on the target domain. The network weights are then updated via backpropagation using the constructed data batch. The additional PoseNet required for unsupervised monocular depth estimation is omitted in this visualization.

artifacts, e.g., overexposure. Similarly, revisiting samples from the source domain addresses the problem of catastrophic forgetting by ensuring that previously acquired knowledge can be preserved. Additionally, the annotations of the source samples enable pseudo-supervision on the target domain by exploiting cross-domain mixing strategies. For both the target and the source replay, we randomly draw multiple samples from the respective replay buffer and apply augmentation to stabilize the loss. In particular, we perform RGB histogram matching of the source images to the online target image, and all available source samples have to be selected once before repetition is allowed to ensure diverse source supervision.

While previous works [17, 32] do not consider limitations on the size of the replay buffer, we explicitly address this challenge to closely resemble the deployment on a robotic platform, where disk storage is an important factor. This poses two questions: First, how to sample from $\mathcal{S}$ to construct the fixed source buffer $\mathbf{B}_\mathcal{S}$ and, second, how to update the dynamic target buffer $\mathbf{B}_\mathcal{T}$? To construct $\mathbf{B}_\mathcal{S}$, we propose a refined version of rare class sampling (RCS) [14]. The frequency f_c of each class $c \in \mathcal{C}$ is calculated based on the number of pixels with class c:

$$f_c = \frac{\sum_{\mathbf{I}\in\mathcal{S}} \sum_{p}^{H\times W} \mathbb{1}_c(p_{c'})}{|\mathcal{S}| \cdot H \cdot W}, \tag{4}$$

where H and W denote the height and width of the images in $\mathcal{S}$ and $p_{c'} \in \mathbf{I}$ refers to a pixel with class c'. The indicator function is 1 if c' equals c and 0 otherwise. The probability of sampling a class is then given by

$$P(c) = \frac{e^{(1-f_c)/T}}{\sum_{c'\in\mathcal{C}} e^{(1-f_{c'})/T}}, \tag{5}$$

with temperature T controlling the smoothness of the distribution, i.e., a smaller T assigns a higher probability to rare classes. In detail, we first sample a class $c \sim P$ and then retrieve all candidate images containing pixels with class c. Instead of taking a random image from these candidates, we sample according to the number of pixels with class c. We repeat both steps $|\mathbf{B}_\mathcal{S}|$ times without selecting the same image more than once. Using RCS ensures that $\mathbf{B}_\mathcal{S}$ contains sufficiently many images with rare classes such that the performance on these classes will further improve during adaptation.

Since $\mathcal{T}$ does not contain annotations and, particularly in the beginning, predictions are not reliable, we cannot use RCS for updating $\mathbf{B}_\mathcal{T}$. Instead, we invert the common methodology of loop closure detection for visual SLAM [32], i.e., the image $\mathbf{I_t}$ is only added to $\mathbf{B}_\mathcal{T}$ if its cosine similarity with respect to all samples within the buffer is below a threshold.

$$\text{sim}_{\cos}(\mathbf{I_t}) = \max_{\mathbf{I}_{\mathcal{T}_\mathbf{i}}\in\mathbf{B}_\mathcal{T}} \cos\left(\text{feat}(\mathbf{I_t}), \text{feat}(\mathbf{I}_{\mathcal{T}_\mathbf{i}})\right), \tag{6}$$

where $\text{feat}(\cdot)$ refers to the image features extracted from the final layer of the shared encoder, which is not adapted. If $\mathbf{B}_\mathcal{T}$ is completely filled, we remove the following image to maximize image diversity:

$$\arg\max_{\mathbf{I}_{\mathcal{T}_\mathbf{i}}\in\mathbf{B}_\mathcal{T}} \sum_{\mathbf{I}_{\mathcal{T}_\mathbf{j}}\in\mathbf{B}_\mathcal{T}} \cos\left(\text{feat}(\mathbf{I}_{\mathcal{T}_\mathbf{i}}), \text{feat}(\mathbf{I}_{\mathcal{T}_\mathbf{j}})\right) \tag{7}$$

Depth Adaptation: To adapt the monocular depth estimation head, we exploit the fact that the photometric error loss (Eq. 1) does not require ground truth annotations. Hence, we can directly transfer it to the implemented continual adaptation. In particular, we compute $\mathcal{L}^d_{pe}$ for the constructed batch $\mathbf{b_t}$ and average the loss such that each sample contributes by the same amount:

$$\mathcal{L}^d_{pe}(\mathbf{b_t}) = \frac{\mathcal{L}^d_{pe}(\mathbf{I_t}) + \sum_i \mathcal{L}^d_{pe}(\mathbf{I}_{\mathcal{T}_\mathbf{i}}) + \sum_j \mathcal{L}^d_{pe}(\mathbf{I}_{\mathcal{S}_\mathbf{j}})}{|\mathbf{b_t}|}. \tag{8}$$

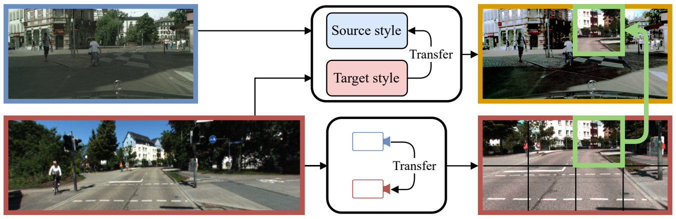

Fig. 3. Our proposed cross-domain mixing strategy first transfers the image style from the target to the source sample. Then it augments the target image to match the appearance of the source camera. Finally, a random image patch is copied from the target to the source image. The source annotations are retained and completed by the network's estimate on the copied image patch, combining self-iterative learning with ground truth supervision.

Furthermore, if the predicted camera motion is below a threshold, i.e., the robot is presumably not moving, we do not compute the $\mathcal{L}^{d}_{pe}(\mathbf{I_t})$ and subtract 1 from the denominator to avoid adding a bias to the current scene.

Panoptic Adaptation: As described in Sec. III-A, panoptic segmentation is the fused output of a semantic head and an instance head. We observe that the decrease in performance on samples from unseen domains can mostly be attributed to the semantic head, while instance predictions remain stable. Cross-domain mixing strategies allow leveraging ideas from supervised training to an unsupervised setting, where ground truth annotations are unknown. In CoDEPS, we bootstrap annotated source samples and high-confident target predictions to artificially generate pseudo-labels for the target samples in an online fashion to supervise the semantic head. Similar to depth adaptation, we continue to compute $\mathcal{L}^{sem}_{bce}$ on $\{\mathbf{I}_{\mathcal{S}_0}, \mathbf{I}_{\mathcal{S}_1}, \dots\}$ to mitigate forgetting, and freeze the instance head.

We further design a mixing strategy combining pixels of images from both $\mathcal{S}$ and $\mathcal{T}$, that considers multiple factors, which are unique to the online continual learning scenario: (1) the robust pretraining on a dedicated source dataset, which may result in significant performance degradation on the target dataset if the pre-trained weights are strongly adapted; (2) the existence of different cameras leading to significant changes in the field-of-view, geometric appearance of objects, resolution, and aspect ratio of the images; and (3) the continuously evolving visual appearance of street scenes during adaptation. To address these challenges, our cross-domain mixing approach employs a three-step method to generate the adaptation signal. First, we perform style transfer from the target image $\mathbf{I}_{\mathcal{T}_\mathrm{i}}$ to the source image $\mathbf{I}_{\mathcal{S}_\mathrm{j}}$ by aligning their pixel value histograms, as depicted in Fig. 3. This allows supervision with ground truth labels on images that are of similar visual appearance as the target image. Second, we apply a geometric transformation on $\mathbf{I}_{\mathcal{T}_\mathrm{i}}$ based on the camera intrinsics of the source and target domains denoted by $\mathbf{K}_{\mathcal{S}}$ and $\mathbf{K}_{\mathcal{T}}$, respectively. To this end, we assume a constant depth distribution over $\mathbf{I}_{\mathcal{S}_\mathrm{j}}$, lift the pixel values into Euclidean space via inverse camera projection, and project the lifted points back into the camera view of $\mathbf{I}_{\mathcal{T}_\mathrm{i}}$ as follows:

$$\mathbf{I}'_{\mathcal{T}} = \mathbf{I}_{\mathcal{T}}\langle \mathbf{K}_{\mathcal{T}}\mathbf{K}_{\mathcal{S}}^{-\mathbf{1}}\mathbf{I}_{\mathcal{S}}\rangle, \tag{9}$$

where $\langle\cdot\rangle$ denotes the bilinear sampling operator. Eq. 9 results in an adapted target image $\mathbf{I}'_{\mathcal{T}}$ with an adjusted field of view, resolution, and a geometric appearance of the scene similar to that of $\mathbf{I}_{\mathcal{S}}$. The final step in the process involves separating $\mathbf{I}'_{\mathcal{T}}$ into multiple segments and randomly selecting one of them to be inserted into the style-transferred source image, see Fig. 3. To avoid providing a flawed supervision signal caused by geometrically unrealistic images, we only insert a single path. Similarly, the ground truth labels of the pixels from $\mathbf{I}_{\mathcal{S}}$ are retained, and the semantic labels estimated by the network are used to label the inserted patch after intrinsics transformation. The generated image is then fed into the network and training is performed using the cross-entropy loss and the generated pseudo-labels of the mixed image. To mitigate the decline in performance commonly associated with self-iterative training on predicted pseudo-labels, often resulting in class collapse, we utilize an exponentially moving average (EMA) filter for updating the network weights. In detail, we create a duplicate with network weights w_{EMA} of the initial model with weights w and use this so-called EMA model to generate the semantic predictions. During continual learning, the weights w are updated via backpropagation on $\mathbf{b_t}$. Then, the EMA model is updated as follows:

$$w_{\text{EMA}} \leftarrow \alpha \cdot w_{\text{EMA}} + (1-\alpha)\cdot w, \tag{10}$$

where α denotes the contribution of the EMA model.

IV. Experimental Evaluation

In the following sections, we provide further details on the pretraining step and the datasets that we evaluate on. We present extensive experimental results on the efficiency and efficacy of our proposed approach and include ablation studies on important design choices. Finally, we expand the experimental setup to multi-domain adaptation closely resembling classical continual learning settings.

We follow the evaluation protocol of Zhang et al. [35]. In detail, we compute the evaluation metrics on the frame of the current timestamp before using the same frame to perform backpropagation and update the model weights. Once 70 % of a sequence is processed, we calculate the average of the accumulated metrics. Additionally, we report the scores on the remaining 30 % of the same sequence without further weight updates to analyze the performance of the adapted model. In the tables, we refer to these types of evaluation by protocol 1 and protocol 2, respectively. We further denote the respective parts of a sequence by *adapt* and *eval*. Unlike Zhang et al. [35], we define our task in the context of continual learning. To measure knowledge retention and hence mitigate catastrophic forgetting, we introduce protocol 3 as evaluating the adapted model on the *val* split of the source dataset.

A. Datasets

To simulate data from a variety of domains, we employ our method on three datasets, namely Cityscapes [7], KITTI-360 [21], and SemKITTI-DVPS [1]. In particular, we utilize Cityscapes for pre-training and sequences of both KITTI-360 and SemKITTI-DVPS for adaptation.

TABLE I
EFFICACY OF THE NETWORK

Method	**Dataset**	**mIoU** ↑	**RMSE** ↓	**Abs Rel** ↓	δ_1 ↑
GUDA	KITTI	—	4.42	0.11	0.88
CoDEPS		62.8	3.52	0.09	0.90
GUDA	Cityscapes	72.9	—	—	—
CoDEPS		72.9	10.16	0.19	0.78

Our utilized network is able to reproduce the performance of the baseline method GUDA [12]. Performance of GUDA is reported by the authors. To evaluate CoDEPS on KITTI, we use sequence 08 *eval* of SemKITTI-DVPS.

TABLE II
ADAPTATION PERFORMANCE

Method	**Sequence**	**mIoU**	**PQ**	**SQ**	**RQ**	**RMSE**	**Abs Rel**
			Protocol 1				
Only target	KITTI-360 seq. 09	68.60	51.00	82.00	59.04	3.49	0.11
Only source		50.59	37.26	74.06	47.38	6.03	0.36
CoDEPS		52.29	38.02	74.88	48.21	4.74	0.19
Only target	KITTI-360 seq. 10	64.65	41.91	76.68	51.48	6.13	0.15
Only source		51.94	32.60	71.27	32.60	8.06	0.35
CoDEPS		53.02	33.50	71.53	33.50	7.19	0.22
			Protocol 2				
Only target	KITTI-360 seq. 09	69.90	48.84	79.36	57.49	3.18	0.10
Only source		50.78	36.57	72.22	46.75	5.60	0.35
CoDEPS		51.53	37.56	72.87	47.99	4.56	0.16
Only target	KITTI-360 seq. 10	55.12	36.58	67.41	46.15	4.78	0.12
Only source		45.74	30.62	69.56	39.49	7.90	0.33
CoDEPS		49.91	31.91	70.68	40.95	5.57	0.15

Comparison with baselines without adaptation evaluated on the *adapt* part (protocol 1) and on the *eval* part (protocol 2) of the specified sequences. “Only target” and “only source” refer to the domain used for training.

Cityscapes: The Cityscapes Dataset [7] is a large-scale autonomous driving dataset that was recorded in 50 cities in Germany and bordering regions. It includes RGB images, panoptic annotations, and vehicle metadata. In this work, we utilize the fine panoptic labels to train the semantic and instance heads in a supervised manner. Additionally, we leverage the sequence image data of the left camera to train the depth prediction in an unsupervised fashion. Finally, we compute the depth error metrics using the provided disparity maps.

KITTI-360: The KITTI-360 Dataset [21] is a relatively recently released public dataset for the domain of autonomous driving, which was recorded in the city of Karlsruhe, Germany. It includes both 2D and 3D panoptic annotations for RGB images and LiDAR data. In this work, we predominantly utilize the RGB images to simulate an online image stream of an onboard camera. In particular, we use these images to adapt our network in a self-supervised manner. To compute evaluation metrics, we compare our predictions with the ground truth measurements and annotations of the dataset.

SemKITTI-DVPS: The SemKITTI-DVPS [29] is based on the odometry benchmark of the KITTI Dataset [8], which was recorded in Karlsruhe, Germany. We utilize the RGB images to simulate an onboard camera and to adapt our network to the new domain. Furthermore, we compute depth metrics based on the provided projected LiDAR points and the semantic/panoptic metrics using the extension SemanticKITTI [1].

Semantic Labels: As the aforementioned datasets use different labeling policies for the semantic annotations, we use the 19 classes of Cityscapes as the reference definition and remap classes of the other datasets. However, certain classes do not exist in the adaptation datasets (*wall*, *traffic light*, *bus*, *train*). For consistency across the datasets, we merge *wall* with *building* and remove the other three classes. Additionally, we merge *motorcycle* and *bicycle* into *two-wheeler* to increase the number of annotated pixels. Consequently, we consider nine “stuff” classes and five “thing” classes, listed in Table V. Note that *sky* is not included in SemKITTI-DVPS due to using LiDAR annotations and hence excluded in the evaluation on this dataset.

B. Pretraining Protocol

The initial state of the network weights before adaptation is obtained by initializing the encoders using pretrained weights from the ImageNet dataset, followed by training the entire model on the Cityscapes dataset. In detail, we use the Adam optimizer with a constant learning rate $lr = 0.0001$ and train the entire network for 250 epochs. In our experiments, we compare the performance of our approach to directly training on the target dataset, which can be considered as a theoretical upper limit having full target knowledge. Due to the unbalanced class distribution of KITTI-360, we train the network in two steps, using the Adam optimizer with $lr = 0.0001$ on sequences 00-07. We train for 45 epochs while ignoring the most common classes *road*, *sidewalk*, *building*, and *vegetation*, followed by 55 epochs including all classes. Similarly, for SemKITTI-DVPS, we train the network on sequences 00-06, 09, and 10 for 30 epochs without the aforementioned classes plus *terrain* and *sky*, which is not included in the dataset, followed by 30 epochs including all classes. In Table I, we demonstrate that our implemented network is able to reproduce the performance of the baseline method GUDA [12].

C. Online Adaptation

In this section, we extensively evaluate our proposed CoDEPS with respect to both adapting to a new domain and retaining knowledge to mitigate forgetting. In detail, for all presented experiments, we freeze the shared encoder following the study by McCraith et al. [25]. Based on the ablation study in Sec. IV-D, we use a buffer size of 300. For RCS, we follow Hoyer et al. [14] and set $T = 0.01$. Updating the EMA model is done with $\alpha = 0.9$.

In Table II, we assess the performance of CoDEPS on sequences 09 and 10 of KITTI-360 and rank it against (theoretical) supervision from either the source or target domain. First, we directly train CoDEPS using available ground truth labels on $\mathcal{T}$ (*only target*). Although this approach is not feasible for real-world deployment, it serves as an upper bound on the potential performance of the network if target knowledge was available. The second experiment involves inference of CoDEPS on $\mathcal{T}$ without adaptation (*only source*) to set the

TABLE III
CONTINUAL LEARNING FOR MONOCULAR DEPTH ESTIMATION

Method	**Batch**	**Protocol 1**					**Protocol 2**					**Protocol 3**				
	current/target/source	RMSE	Abs Rel	δ_1	δ_2	δ_3	RMSE	Abs Rel	δ_1	δ_2	δ_3	RMSE	Abs Rel	δ_1	δ_2	δ_3
Only source	0 / 0 / 0	8.06	0.35	0.43	0.77	0.91	7.90	0.33	0.44	0.77	0.93	**10.16**	**0.19**	**0.78**	**0.93**	**0.97**
Online image	1 / 0 / 0	8.33	0.27	0.64	0.84	0.93	6.06	0.33	0.46	0.73	0.90	13.72	0.57	0.30	0.50	0.68
Target replay	1 / 2 / 0	**6.35**	**0.19**	**0.77**	**0.91**	**0.96**	**5.34**	**0.15**	**0.81**	**0.93**	**0.97**	12.48	0.44	0.34	0.68	0.88
CoDEPS	1 / 2 / 2	7.19	0.22	0.73	0.89	0.94	5.57	**0.15**	**0.81**	**0.93**	**0.97**	11.38	0.21	0.75	0.91	0.96

The root mean squared error (RMSE), absolute relative error (Abs Rel) as well as accuracies $\delta_1 = \delta < 1.25$, $\delta_2 = \delta < 1.25^2$, and $\delta_3 = \delta < 1.25^3$, obtained by adapting Cityscapes to sequence 10 of the KITTI-360 dataset. Best results in each category are in **bold**; second best are underlined.

TABLE IV
CONTINUAL LEARNING FOR PANOPTIC SEGMENTATION

Method	**Protocol 1**				**Protocol 2**				**Protocol 3**			
	mIoU	PQ	SQ	RQ	mIoU	PQ	SQ	RQ	mIoU	PQ	SQ	RQ
Only source	51.94	32.60	71.27	42.44	45.74	30.62	69.56	39.49	72.87	49.19	77.45	60.40
GUDA [12]	45.56	29.70	70.67	39.05	47.62	31.03	64.00	40.49	66.57	44.39	75.95	55.32
DACS [30]	51.14	32.09	71.12	42.23	45.24	29.05	69.47	38.11	72.66	49.27	77.33	60.60
CoDEPS (online image)	**53.22**	33.46	**71.63**	43.46	49.51	31.49	64.17	40.71	72.81	**49.83**	77.25	**61.49**
CoDEPS (random sampling)	52.36	33.24	71.60	43.25	48.78	31.50	68.83	40.56	72.05	49.11	77.18	60.52
CoDEPS	53.02	**33.50**	71.53	**43.62**	**49.91**	**31.91**	**70.68**	**40.95**	**72.90**	49.76	**77.49**	61.22

The mean intersection over union (mIoU), panoptic quality (PQ), semantic quality (SQ), and recognition quality (RQ) are obtained by adapting Cityscapes to sequence 10 of the KITTI-360 dataset. Best results in each category are in **bold**; second best are underlined.

lower performance bound. Finally, the third experiment refers to adapting the network using our proposed CoDEPS with the online stream of unlabeled data from $\mathcal{T}$. We demonstrate the key performance metrics of both protocols 1 and 2. Table II shows that CoDEPS achieves a significant performance boost, as measured by the mIoU metric and all depth metrics. We attribute this improvement to the additional supervision signals incorporated into the segmentation head through our mixing strategy and the self-supervised reconstruction loss for depth adaptation. The improvement in semantic segmentation further enhances the panoptic segmentation metrics.

For the following experiments, we consider the case of using Cityscapes as the source domain and sequence 10 of KITTI-360 as the target domain. In Fig. 5, we illustrate the adaptation progress using unseen validation samples and compare the results to the ground truth. For depth, we visualize predictions generated by the network if it was only trained on $\mathcal{S}$ and $\mathcal{T}$, respectively. For panoptic segmentation, the progressive adaptation on the target domain is particularly visible on the *sidewalk* and *terrain* image regions. Furthermore, instances become more pronounced, e.g., the car in the center of the right sample. Nonetheless, CoDEPS successfully maintains its performance on the source domain with only minimum decreases in depth estimationl

Depth Adaptation: We present the results for monocular depth estimation in Table III. The first row shows the performance on $\mathcal{T}$ (protocols 1 and 2) without adaptation. Comparing the absolute relative error as well as the accuracies δ_1, δ_2, and δ_3 to the performance on $\mathcal{S}$ reveals the domain gap. While continual learning using the current online sample increases the accuracy of protocol 1, it also overfits to the current scene. That is, generalizability to the entire target domain is not achieved as shown by protocol 2. Introducing replay samples from the target buffer overcomes this issue and accounts for online samples of poor quality, improving protocols 1 and 2. However, both of the above result in catastrophic forgetting with respect to $\mathcal{S}$ (protocol 3). The final CoDEPS adds additional source replay yielding low errors and high accuracy by compromising on both $\mathcal{S}$ and $\mathcal{T}$.

Panoptic Adaptation: We further evaluate CoDEPS by comparing it with two competitive baselines that perform domain adaptation on segmentation tasks: GUDA [12], which combines semantic segmentation and depth estimation, rendering their task comparable to ours, and DACS [30], which employs a class-mix strategy for offline domain adaptation of semantic segmentation. To ensure a fair comparison, both baselines are evaluated using the same settings as CoDEPS, including diversity sampling-based experience replay. The results in Table IV indicate that both approaches lead to a significant performance decrease across all three protocols. GUDA's reliance on self-supervised feature alignment using depth training is not effective in the continual learning setting, as shown in the results. DACS also suffers from a decline in performance, likely due to the strong intervention of its mixing strategy into the pretrained network, which can already produce reasonable predictions on the target domain without adaptation.

These results imply that traditional approaches from offline sim-to-real adaptation may not perform well in the online continual learning scenario. To further assess the impact of target replay and our diversity-based buffer sampling, we selectively deactivate both components. Applying the proposed cross-domain mixing strategy results in an improvement in

TABLE V
CLASSWISE EVALUATION

	Class	Only target	Only source	CoDEPS
Stuff	Road	93	89	91
	Sidewalk	40	32	37
	Building	88	85	85
	Fence	43	14	22
	Pole	35	29	32
	Traffic sign	40	35	38
	Vegetation	78	73	75
	Terrain	54	21	39
	Sky	82	79	81
Thing	Person	47	38	38
	Rider	47	29	36
	Car	91	83	84
	Truck	1	4	2
	Two-wheeler	33	27	38
	Mean	55.1	45.7	49.9

The classwise mIoU is based on protocol 2 in Table IV. We compare CoDEPS against two baselines that were trained using source ("only source") or target data ("only target""), respectively. CoDEPS provides a significant performance boost of 4.2% in terms of the mIoU metric.

protocol 1. However, similar to depth adaptation, the results are not fully generalizable to the entire target domain, e.g., SQ of protocol 2. Instead of diversity-based sampling, we use random sampling when both creating the source buffer and when updating the target buffer. Compared to CoDEPS, the performance heavily degrades demonstrating the efficacy of the sampling method. Finally, we present the classwise evaluation of the segmentation performance in Table V, which demonstrates improvements of CoDEPS in the IoU metrics for most classes. In particular, we observe significant enhancements of the *two-wheeler* and *terrain* classes. The latter can also be observed in Fig. 5. In fact, CoDEPS outperforms even the model trained directly on the target domain using ground truth supervision for the latter class.

D. Ablation Study of the Replay Buffer

We extensively study different sizes of the replay buffer and the effect of diversity sampling as explained in Sec. III-B. We list our results in Table VI. Note that an infinite replay buffer contains 2,975 source and a maximum of 2,683 target samples in the employed setting, i.e., adapting from Cityscapes *train* to KITTI-360 using sequence 10 *adapt*. Generally, a larger replay buffer yields higher performance with respect to both adaptation capability and avoiding catastrophic forgetting. Additionally, the proposed diversity sampling using semantic classes for the source and image features for the target samples increases the performance throughout the experiments. However, a greater buffer size increases the required storage posing a challenge for real-world deployment. Based on the presented results, we select a buffer size of 300 with active diversity sampling as for smaller sizes the performance of semantic segmentation on the target domain degrades.

E. Continual Adaptation

Finally, we evaluate the performance of CoDEPS in the context of multi-domain adaptation, i.e., $\mathcal{S} \rightarrow \mathcal{T}_1 \rightarrow \mathcal{T}_2$. In particular, we first adapt to sequence 10 of KITTI-360 followed by sequence 08 of SemKITTI-DVPS, then we invert the adaptation order. To analyze forward and backward transfer as defined for continual learning [22], we compute the metrics on the *val* split of the source and the *adapt* parts of the respective target domains. We report the results in Table VII. Note that we use $\alpha_{\mathcal{S}\rightarrow\mathcal{T}_1} = 0.9$ and $\alpha_{\mathcal{T}_1\rightarrow\mathcal{T}_2} = 0.7$ for updating the EMA model according to Eq. 10 since the network should adapt more strongly when deployed to $\mathcal{T}_2$ due to the larger amount of previously seen data. As shown in the first row of both adaptation orders, CoDEPS is able to mitigate catastrophic forgetting with respect to $\mathcal{S}$ maintaining its performance. We make a similar observation when re-evaluating $\mathcal{T}_1$ after the second adaptation step to $\mathcal{T}_2$. In particular, CoDEPS achieves positive backward transfer on SemKITTI-DVPS when adapting to KITTI-360. On the same adaptation order, we observe positive forward transfer for KITTI-360, i.e., the performance increases although CoDEPS was only adapted to SemKITTI-DVPS.

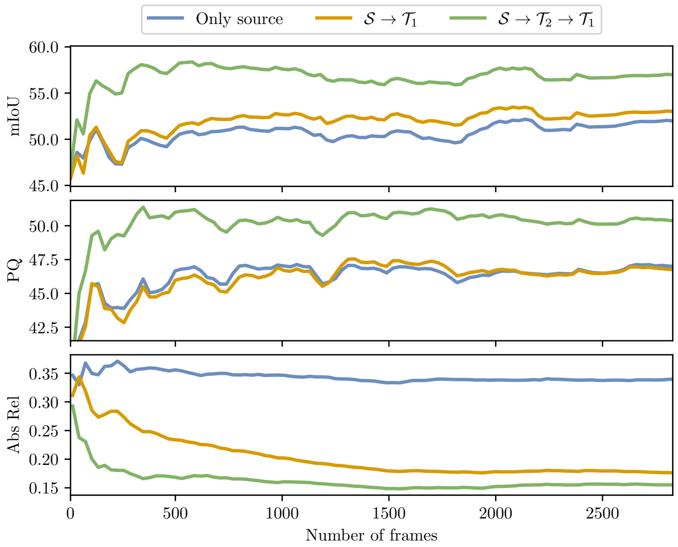

Fig. 4. Evolution of performance metrics on SemKITTI-DVPS sequence 08 during adaptation (protocol 1). The metrics are averaged until the given frame number. The target domains $\mathcal{T}_1$ and $\mathcal{T}_2$ refer to SemKITTI-DVPS and KITTI-360, respectively. It can be seen that there is positive forward transfer when first adapting on $\mathcal{T}_2$.

In Fig. 4, we illustrate the evolution of the performance metrics on SemKITTI-DVPS sequence 08 during adaptation (protocol 1). We compare the error without adaptation to directly adapting to SemKITTI-DVPS versus first adapting to KITTI-360. For both semantic segmentation and depth estimation, it can be clearly observed that the performance improves if more images have been seen. Additionally, adapting first to KITTI-360 results in a large performance increase for both semantic and panoptic segmentation. We account this to the fact that KITTI-360 sequence 10 leads to strongly improved performance, shown in Table VII, that can be transferred to the SemKITTI-DVPS domain.

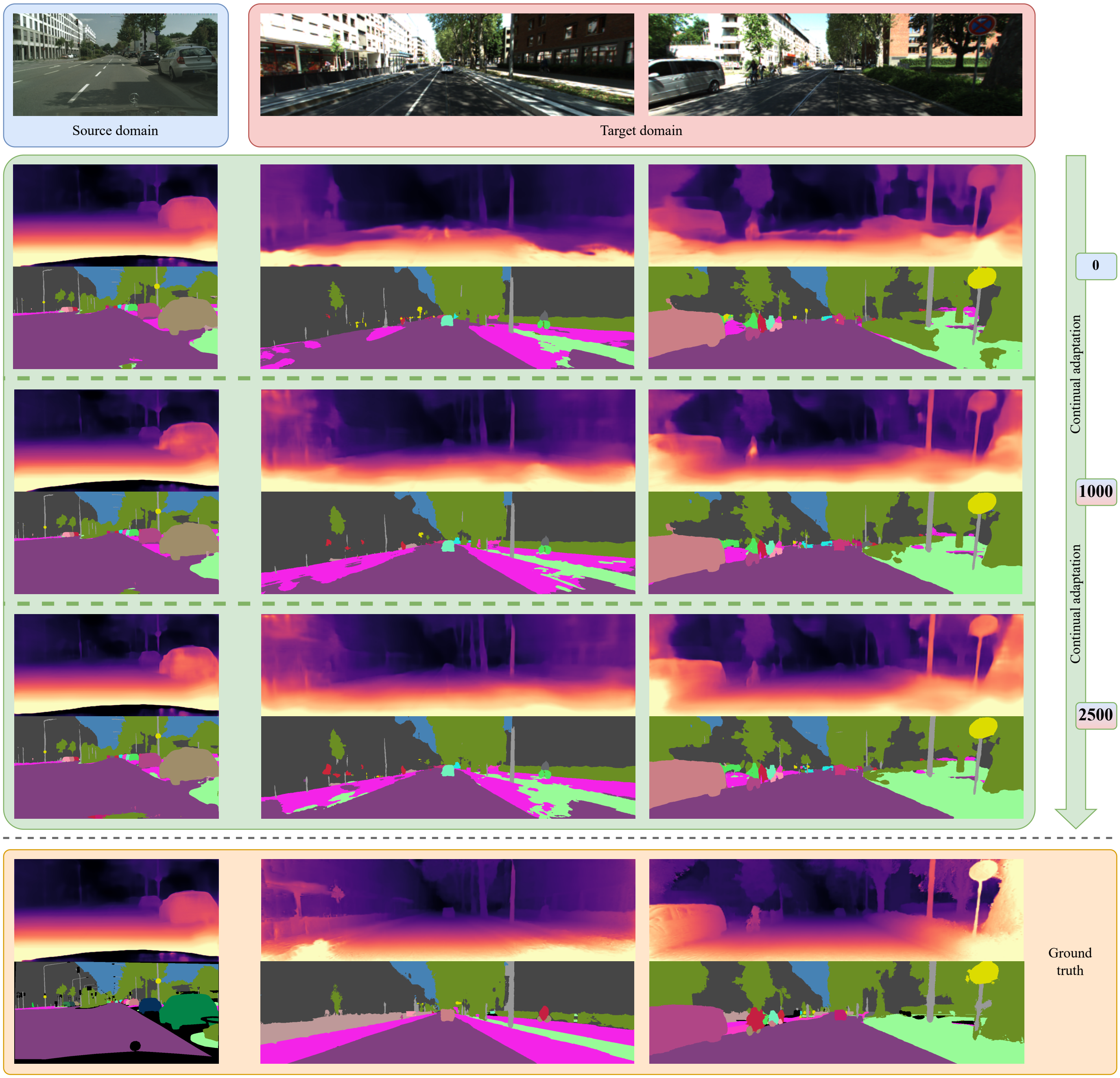

Fig. 5. Qualitative results for Cityscapes to KITTI-360 adaptation after pretraining on the source, i.e., 0 steps, and after having seen 1,000 and 2,500 frames. As shown in the left column, CoDEPS is able to avoid catastrophic forgetting on the source domain. The progressive adaptation on the target domain is particularly visible on the *sidewalk* (left) and *terrain* (right) image regions. Furthermore, instances become more pronounced, e.g., see the car in the center of the right sample.

TABLE VI
ABLATION STUDY ON THE REPLAY BUFFER

Size	Div.	Protocol 2						Protocol 3					
		mIoU	PQ	SQ	RQ	RMSE	Abs Rel	mIoU	PQ	SQ	RQ	RMSE	Abs Rel
∞		49.15	31.95	69.08	40.96	4.94	0.15	73.25	50.37	77.77	61.87	10.76	0.21
1000		49.11±0.69	31.85±0.25	66.82±3.06	40.93±0.06	**5.04**±0.01	**0.14**±0.00	72.84±0.33	49.93±0.20	77.51±0.05	61.39±0.28	11.35±0.39	0.22±0.01
1000	✓	49.36	31.83	68.89	**41.01**	5.30	0.15	**73.50**	**50.05**	**77.67**	**61.48**	12.06	0.23
500		48.77±0.39	31.54±0.39	67.39±2.16	40.66±0.54	5.20±0.20	0.15±0.00	72.38±0.26	49.48±0.14	77.45±0.16	60.90±0.23	**11.14**±0.54	0.22±0.01
500	✓	49.56	31.83	70.11	40.96	5.55	0.16	72.78	49.68	77.39	61.10	11.30	0.22
300		48.78±0.05	31.50±0.19	68.83±2.31	40.56±0.15	5.27±0.16	0.15±0.00	72.05±0.25	49.11±0.30	77.18±0.07	60.52±0.35	**11.14**±0.22	0.22±0.01
300	✓	**49.91**	**31.91**	**70.68**	40.95	5.57	0.15	72.90	49.76	77.49	61.22	11.38	**0.21**
100		48.27±0.84	30.71±0.41	63.95±0.41	39.79±0.38	5.83±0.15	0.16±0.00	69.75±1.77	47.94±0.95	76.66±0.25	59.39±1.16	10.86±0.56	0.22±0.02
100	✓	48.40	30.85	64.07	39.95	5.31	0.16	72.35	48.81	77.16	60.25	11.71	0.22
25		46.03±1.03	29.62±0.37	66.10±2.26	38.48±0.45	5.25±0.26	**0.14**±0.01	67.23±0.85	45.90±0.66	75.69±0.38	57.21±0.76	11.81±0.22	0.22±0.01
25	✓	46.35	29.73	63.35	38.58	5.62	0.17	68.84	46.34	76.06	57.78	12.51	0.24

The numbers above are obtained by adapting Cityscapes to sequence 10 of the KITTI-360 dataset. Here, an infinite buffer size equals 2,975 source samples and a maximum of 2,683 target samples. Note that the effective size is two times the shown value as it refers to both source and target replay. The term "Div." refers to diversity sampling. Where diversity sampling is not used, the same experiment is repeated three times with different random seeds to ensure a statistically reliable measure of performance. The results of these experiments are presented as the mean and standard deviation. Best results in each category are in **bold**; second best are underlined.

TABLE VII
CONTINUAL LEARNING ON MULTIPLE DOMAINS

Domain	**mIoU**	**PQ**	**SQ**	**RQ**	**RMSE**	**Abs Rel**	**mIoU**	**PQ**	**SQ**	**RQ**	**RMSE**	**Abs Rel**	**mIoU**	**PQ**	**SQ**	**RQ**	**RMSE**	**Abs Rel**
	⟶ Pretraining on Cityscapes ⟶						Adaptation on KITTI-360 ⟶						Adaptation on SemKITTI-DVPS ⟶					
Cityscapes	72.87	49.19	77.45	60.40	10.16	0.19	72.90	49.76	77.49	61.22	11.38	0.21	72.42	48.74	77.08	60.20	10.65	0.21
KITTI-360 seq. 10	45.74	30.62	69.56	39.49	7.90	0.33	49.91	31.91	70.68	40.95	5.57	0.15	49.26	32.32	64.08	40.95	5.23	0.15
SemKITTI-DVPS seq. 08	51.95	45.24	76.07	57.20	6.17	0.34	49.48	43.26	74.24	57.26	5.60	0.21	53.70	46.50	76.53	59.43	4.32	0.16
	⟶ Pretraining on Cityscapes ⟶						Adaptation on SemKITTI-DVPS ⟶						Adaptation on KITTI-360 ⟶					
Cityscapes	72.87	49.19	77.45	60.40	10.16	0.19	72.75	49.01	77.36	60.35	10.82	0.22	72.51	48.87	76.98	60.28	11.41	0.21
KITTI-360 seq. 10	45.74	30.62	69.56	39.49	7.90	0.33	49.26	31.66	70.26	41.40	6.30	0.17	50.05	31.92	70.50	41.48	5.47	0.16
SemKITTI-DVPS seq. 08	51.95	45.24	76.07	57.20	6.17	0.34	52.31	44.29	75.58	56.87	4.56	0.16	53.83	47.29	76.55	60.01	4.25	0.16

CoDEPS is continually applied to three domains using Cityscapes as the initial source domain and then adapting to KITTI-360 and SemKITTI-DVPS. The listed numbers on the target domains are based on protocol 2.

V. CONCLUSION

In this paper, we present CoDEPS as the first approach for online continual learning for joint monocular depth estimation and panoptic segmentation. CoDEPS enables the vision system of a robotic platform to continually enhance its performance in an online fashion. In particular, we propose a new cross-domain mixing strategy to adapt panoptic segmentation combining annotated source data with unlabeled images from a target domain. To mitigate catastrophic forgetting, CoDEPS leverages experience replay using a buffer composed of source and target samples. Unlike prior works, we explicitly address the limited memory capacity of robotic platforms by setting a fixed size for the replay buffer. To ensure distinct replay samples, we use rare class sampling on the source set and employ image-based diversity sampling when updating the target buffer. Using extensive evaluations, we demonstrate that CoDEPS outperforms competitive baselines while avoiding catastrophic forgetting in the online continual learning setting. Future work will explore cross-task synergies and the use of pretext tasks for domain adaptation.

ACKNOWLEDGMENT

This work was partly funded by the European Union's Horizon 2020 research and innovation program under grant agreement No 871449-OpenDR and the Bundesministerium für Bildung und Forschung (BMBF) under grant agreement No FKZ 16ME0027.

REFERENCES

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. In *International Conference on Computer Vision*, 2019.

[2] Borna Bešić and Abhinav Valada. Dynamic object removal and spatio-temporal RGB-D inpainting via geometry-aware adversarial learning. *IEEE Transactions on Intelligent Vehicles*, 7(2):170–185, 2022.

[3] Borna Bešić, Nikhil Gosala, Daniele Cattaneo, and Abhinav Valada. Unsupervised domain adaptation for LiDAR panoptic segmentation. *IEEE Robotics and Automation Letters*, 7(2):3404–3411, 2022.

[4] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Unsupervised monocular depth and ego-motion learning with structure and semantics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[5] Bowen Cheng, Maxwell D. Collins, Yukun Zhu, Ting Liu, Thomas S. Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In

IEEE/CVF Conference Computer Vision and Pattern Recognition, pages 12472–12482, 2020.

[6] Gong Cheng and James H. Elder. VCSeg: Virtual camera adaptation for road segmentation. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1969–1978, 2022.

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *IEEE/CVF Conference Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *IEEE/CVF Conference Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[9] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *IEEE/CVF Conference Computer Vision and Pattern Recognition*, pages 270–279, 2017.

[10] Clement Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. In *International Conference on Computer Vision*, pages 3827–3837, 2019.

[11] Nikhil Gosala and Abhinav Valada. Bird's-eye-view panoptic segmentation using monocular frontal view images. *IEEE Robotics and Automation Letters*, 7(2):1968–1975, 2022.

[12] Vitor Guizilini, Jie Li, Rareș Ambruș, and Adrien Gaidon. Geometric unsupervised domain adaptation for semantic segmentation. In *International Conference on Computer Vision*, pages 8537–8547, 2021.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[14] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. DAFormer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *IEEE/CVF Conference Computer Vision and Pattern Recognition*, pages 9924–9935, 2022.

[15] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Cross-view regularization for domain adaptive panoptic segmentation. In *IEEE/CVF Conference Computer Vision and Pattern Recognition*, pages 10133–10144, 2021.

[16] Marvin Klingner, Mouadh Ayache, and Tim Fingscheidt. Continual batchnorm adaptation (CBNA) for semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):20899–20911, 2022.

[17] Yevhen Kuznietsov, Marc Proesmans, and Luc Van Gool. CoMoDA: Continuous monocular depth adaptation using past experiences. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2907–2917, 2021.

[18] Yevhen Kuznietsov, Marc Proesmans, and Luc Van Gool. Towards unsupervised online domain adaptation for semantic segmentation. In *European Conference on Computer Vision*, pages 261–271, 2022.

[19] Hanhan Li, Ariel Gordon, Hang Zhao, Vincent Casser, and Anelia Angelova. Unsupervised monocular depth learning in dynamic scenes. In *Conference on Robot Learning*, pages 1908–1917. PMLR, 2021.

[20] Shunkai Li, Xin Wang, Yingdian Cao, Fei Xue, Zike Yan, and Hongbin Zha. Self-supervised deep visual odometry with online adaptation. In *IEEE/CVF Conference Computer Vision and Pattern Recognition*, pages 6339–6348, 2020.

[21] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2022.

[22] David Lopez-Paz and Marc' Aurelio Ranzato. Gradient episodic memory for continual learning. In *Conference on Neural Information Processing Systems*, volume 30, 2017.

[23] Adrian Lopez-Rodriguez and Krystian Mikolajczyk. DESC: Domain adaptation for depth estimation via semantic consistency. *arXiv preprint arXiv:2009.01579*, 2020.

[24] Giulio Mattolin, Luca Zanella, Elisa Ricci, and Yiming Wang. ConfMix: Unsupervised domain adaptation for object detection via confidence-based mixing. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 423–433, January 2023.

[25] Robert McCraith, Lukas Neumann, Andrew Zisserman, and Andrea Vedaldi. Monocular depth estimation with self-supervised instance adaptation. *arXiv preprint arXiv:2004.05821*, 2020.

[26] Rohit Mohan and Abhinav Valada. Amodal panoptic segmentation. In *IEEE/CVF Conference Computer Vision and Pattern Recognition*, pages 20991–21000, 2022.

[27] Rohit Mohan and Abhinav Valada. Perceiving the invisible: Proposal-free amodal panoptic segmentation. *IEEE Robotics and Automation Letters*, 7(4):9302–9309, 2022.

[28] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. ClassMix: Segmentation-based data augmentation for semi-supervised learning. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1368–1377, 2021.

[29] Siyuan Qiao, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. ViP-DeepLab: Learning visual perception with depth-aware video panoptic segmentation. In *IEEE/CVF Conference Computer Vision and Pattern Recognition*, pages 3996–4007, 2021.

[30] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. DACS: Domain adaptation via cross-domain mixed sampling. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1378–1388, 2021.

[31] Abhinav Valada, Gabriel Oliveira, Thomas Brox, and Wolfram Burgard. Towards robust semantic segmentation using deep fusion. In *Robotics: Science and Systems (RSS*

2016) Workshop, Are the Sceptics Right, 2016.

[32] Niclas Vödisch, Daniele Cattaneo, Wolfram Burgard, and Abhinav Valada. Continual SLAM: Beyond lifelong simultaneous localization and mapping through continual learning. In Aude Billard, Tamim Asfour, and Oussama Khatib, editors, *Robotics Research*, pages 19–35, Cham, 2023. Springer Nature Switzerland.

[33] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *IEEE/CVF Conference Computer Vision and Pattern Recognition*, pages 7201–7211, 2022.

[34] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3VO: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *IEEE/CVF Conference Computer Vision and Pattern Recognition*, pages 1281–1292, 2020.

[35] Zhenyu Zhang, Stéphane Lathuilière, Elisa Ricci, Nicu Sebe, Yan Yan, and Jian Yang. Online depth learning against forgetting in monocular videos. In *IEEE/CVF Conference Computer Vision and Pattern Recognition*, pages 4493–4502, 2020.

[36] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and egomotion from video. In *IEEE/CVF Conference Computer Vision and Pattern Recognition*, pages 1851–1858, 2017.