CoVIO: Online Continual Learning for Visual-Inertial Odometry

Niclas Vödisch[1]   Daniele Cattaneo[1]   Wolfram Burgard[2]   Abhinav Valada[1]

[1]University of Freiburg   [2]University of Technology Nuremberg

Abstract

Visual odometry is a fundamental task for many applications on mobile devices and robotic platforms. Since such applications are oftentimes not limited to predefined target domains and learning-based vision systems are known to generalize poorly to unseen environments, methods for continual adaptation during inference time are of significant interest. In this work, we introduce CoVIO for online continual learning of visual-inertial odometry. CoVIO effectively adapts to new domains while mitigating catastrophic forgetting by exploiting experience replay. In particular, we propose a novel sampling strategy to maximize image diversity in a fixed-size replay buffer that targets the limited storage capacity of embedded devices. We further provide an asynchronous version that decouples the odometry estimation from the network weight update step enabling continuous inference in real time. We extensively evaluate CoVIO on various real-world datasets demonstrating that it successfully adapts to new domains while outperforming previous methods. The code of our work is publicly available at http://continual-slam.cs.uni-freiburg.de.

1. Introduction

Reliable estimation of a robot's motion based on its on-board sensors is a fundamental requirement for many downstream tasks including localization and navigation. Devices such as inertial measurement units (IMU) or inertial navigation systems (INS) provide a way to directly measure the robot's motion based on acceleration and GNSS readings. An alternative is to use visual odometry (VO) leveraging image data from monocular or stereo cameras. Such VO methods have been successfully used in UAVs [4], mobile applications [24], and even mars rovers [21]. Similar to other vision tasks, learning-based VO has gained increasing attention as the learnable high-level features can circumvent problems in textureless regions [27, 28] or in the presence of dynamic objects [1] where classical handcrafted methods suffer. However, learning-based VO lacks the ability to generalize to unseen domains, hindering their open-world deployment. Recently, adaptive VO [18] has opened a new
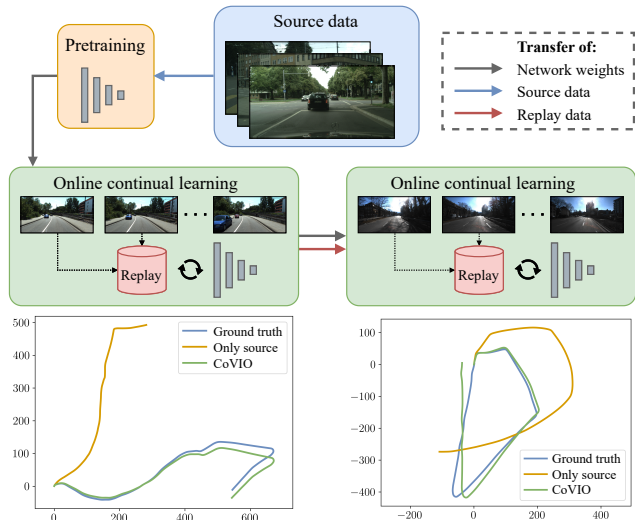

Figure 1. We propose CoVIO for online continual learning of visual-inertial odometry. After pretraining on a source domain that is then discarded, CoVIO further updates the network weights during inference on a target domain. Using experience replay CoVIO successfully mitigates catastrophic forgetting.

avenue of research, *e.g.*, by using continual learning (CL) methodologies to enhance VO during inference time [30].

Most commonly, learning-based VO leverages monocular depth estimation as an auxiliary task [13, 18, 30] and exploits an unsupervised joint training scheme of a PoseNet, estimating the camera motion between two frames, and a DepthNet, estimating depth from a single image [6]. Due to the unsupervised nature of this approach, learning-based VO can be continuously trained also during inference time. In addition to classical domain adaptation [2], where knowledge is transferred from a single source to a single target domain, the recent study on continual SLAM [30] also investigates a sequential multi-domain setting as illustrated in Fig. 1. The authors introduce CL-SLAM, which fuses adaptive VO with a graph-based SLAM backend. To avoid catastrophic forgetting, *i.e.*, overfitting to the current domain while losing the ability to perform well on past domains, CL-SLAM employs a dual-network architecture comprising an expert and a generalizer for both efficient domain

adaptation and knowledge retention combined with experience replay. However, the previously proposed CL-SLAM suffers from three main drawbacks: First, network weights are transferred from the generalizer to the expert upon the start of a new evaluation sequence, *i.e.*, a human supervisor decides when new data should be classified as a domain change. Second, the utilized replay buffer of the generalizer is of infinite size and, thus, does not consider the limited storage capacity of real-world applications. Finally, since every received frame triggers an update of the network weights before yielding the VO estimate, real-time usage is difficult to achieve on low-power devices such as embedded hardware in robots.

In this work, we propose a novel adaptive visual-inertial odometry estimation method called CoVIO that explicitly addresses all of the aforementioned drawbacks of CL-SLAM. Similar to Kuznietsov *et al.* [11], we consider a source-free setting, *i.e.*, experience replay does not include data from the source domain used for pretraining. In particular, the contributions of this work can be summarized as follows:

1. We replace the dual-network architecture with a single network addressing both domain adaptation and knowledge retention but simplifying the overall architecture and reducing the GPU memory footprint. Additionally, this resolves the issue of transferring network weights without domain classification.

2. We propose a fixed-size replay buffer that maximizes image diversity and addresses the limited storage capacity of embedded devices.

3. We present an asynchronous version of CoVIO that separates the core motion estimation from the network update step allowing true continuous inference.

4. We perform extensive evaluations of CoVIO on various datasets, both publicly available and in-house, demonstrating its efficacy compared to other visual odometry methods.

5. We release the code of our work and trained models at http://continual-slam.cs.uni-freiburg.de.

2. Related Work

In this section, we provide a brief introduction to continual and lifelong learning and summarize previous methods for domain adaptation of learning-based visual odometry.

Continual Learning: Deep learning-based models are commonly trained for a specific task, which is defined a priori, using a fixed set of training data. During inference, the model is then employed on previously unseen data from the same domain without further updates of the network weights. However, in many real-world scenarios, this assumption does not hold true, *e.g.*, the initially used training data might not well represent the data seen during inference,

thus leading to a domain gap and suboptimal performance. Additionally, the objective of the task can change over time. Continual learning (CL) and lifelong learning [26] aim to overcome these challenges by enabling a method to continually learn additional tasks given new training data. In contrast to vanilla domain adaptation [2], CL methods should maintain the capability to solve previously learned tasks, *i.e.*, avoiding catastrophic forgetting. Ideally, learning a task also yields improved performance on previous tasks (*positive backward transfer*) as well as on yet unknown future tasks (*positive forward transfer*) [16]. The majority of CL approaches can be categorized into three strategies. First, experience replay directly tackles catastrophic forgetting from a data-driven perspective. For instance, both CoDEPS [31] and CoMoDA [10] store images in a replay buffer and combine online data with replay samples when updating the network weights. Second, regularization techniques such as knowledge distillation [29] preserve information on a more abstract feature level. Finally, architectural methods prevent forgetting by using certain network structures, *e.g.*, LSTMs [13] and dual-network architectures [30], or by directly freezing internal model parameters. Online continual learning [17, 32] describes an extension of CL by considering a setting, where the model is continuously updated on a stream of data during inference time. Online CL also includes scenarios, which gradually change from one domain to another [25]. In this work, we employ online CL with experience replay for learning-based visual-inertial odometry estimation.

Adaptive Visual Odometry: Online adaptation of learning-based visual odometry (VO) and simultaneous localization and mapping (SLAM) aims to enhance performance on the fly allowing robotic systems to operate more reliably in previously unseen environments. Most commonly, learning-based VO relies on monocular depth estimation as an auxiliary task enabling joint training of a DepthNet and a PoseNet in an unsupervised manner [6]. In an early work on adaptive VO, Luo *et al.* [18] accumulate images from an online camera stream and leverage the unsupervised training scheme to update both networks. Different to experience replay in CL, the buffer of accumulated images is emptied after the update step, *i.e.*, each sample is only seen once. Li *et al.* [13] propose an architectural CL technique, replacing the standard convolutional layers with LSTM variants to prevent forgetting. During deployment, the networks are continuously trained using only the online data. In a follow-up work by the same authors [14], the PoseNet is substituted with optical flow-based point matching. Similarly, GeoRefine [9] combines online depth refinement with dense visual mapping. While the DepthNet is updated following the aforementioned works, GeoRefine uses a non-adaptive odometry and tracking module based on optical flow. Loo *et al.* [15] propose an adaptive visual SLAM system that com-
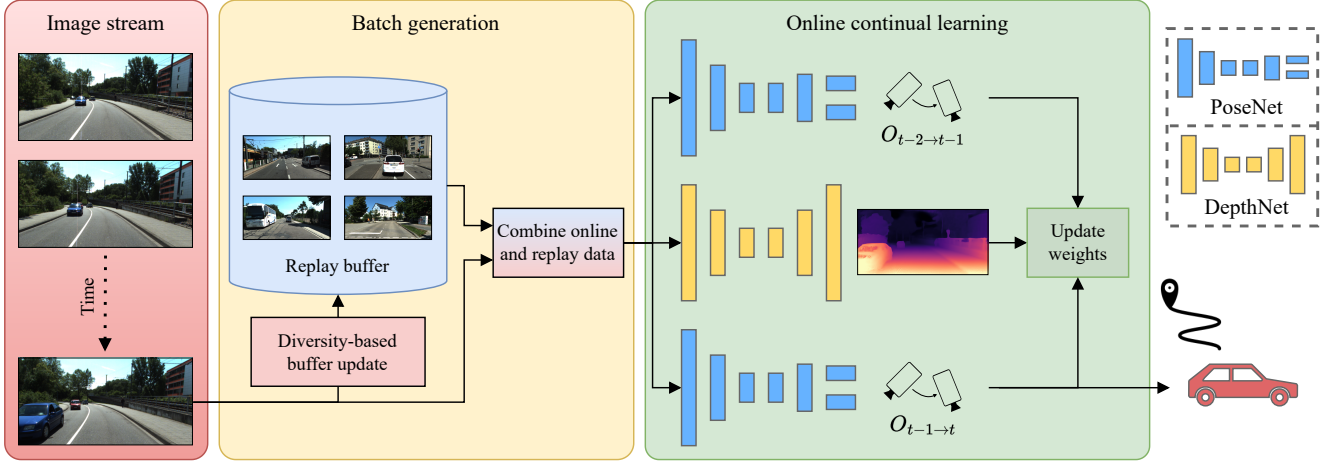
Figure 2. Our proposed CoVIO performs online continual learning on a stream of RGB images leveraging unsupervised monocular depth estimation as an auxiliary task. In each update step, the image triplet consisting of the current and the two previous frames is combined with samples from a replay buffer and then fed to the networks to update their weights via backpropagation. The estimated camera motion between the previous and the current image corresponds to the generated VO output. The replay buffer is optionally updated if the current frame is sufficiently different from the existing content.

bines experience replay with a variant of elastic weight consolidation (EWC) to further regularize the weight updates of both the DepthNet and the PoseNet. Finally, to avoid catastrophic forgetting in a multi-domain adaptation setting, CL-SLAM [30] exploits a dual-network architecture, which is composed of an expert to perform effective online adaptation to the new domain and a generalizer to retain previously acquired knowledge by leveraging experience replay. In this work, we propose an adaptive method for visual-inertial odometry built on CL-SLAM that explicitly addresses its shortcomings as outlined in Sec. 1.

3. Technical Approach

In the following sections, we first describe the network architecture along with the pretraining procedure on a source domain. Then, we introduce CoVIO and provide detailed explanations of all contributions.

3.1. Network Architecture and Pretraining

In this section, we detail the network architecture of our proposed CoVIO and the loss functions that we employ during the initial training phase.

Network Architecture: We build our network following the common scheme of unsupervised monocular depth estimation leveraging two separate networks that we refer to as DepthNet and PoseNet as depicted in Fig. 2. Similar to CL-SLAM [30], for an image triplet $\{\mathbf{I}_{t-2}, \mathbf{I}_{t-1}, \mathbf{I}_t\}$ we use Monodepth2 [6] to jointly predict a dense depth map $\mathbf{D}_{t-1}$ of the center image and the camera motion with respect to both neighboring frames, *i.e.*, $\mathbf{O}_{t-2 \to t-1}$ and $\mathbf{O}_{t-1 \to t}$. In CoVIO, we then output the latter as the VO estimate. In par-

ticular, we use an implementation comprising two separate ResNet-18 [8] encoders for the DepthNet and the PoseNet.

Source Domain Pretraining: To initialize CoVIO, we perform unsupervised training on a source domain $\mathcal{S}$ in an offline manner. In detail, we exploit the photometric reprojection loss $\mathcal{L}_{pr}$ and the image smoothness loss $\mathcal{L}_{sm}$ to train the DepthNet and the PoseNet [6]. We additionally supervise the PoseNet with scalar velocity readings from the vehicle's IMU [7]. The applied velocity supervision term $\mathcal{L}_{vel}$ enforces metric scale-aware odometry estimates. Thus, our total loss is composed of three terms:

$$\mathcal{L} = \mathcal{L}_{pr} + \gamma \mathcal{L}_{sm} + \lambda \mathcal{L}_{vel}, \tag{1}$$

with weighting factors γ and λ.

3.2. Online Continual Learning

After pretraining on a source domain $\mathcal{S}$, we use CoVIO to perform online continual learning on an unseen target domain $\mathcal{T}$. As illustrated in Fig. 2, each new RGB image triggers the following steps:

(1) Create a data triplet comprising the new frame $\mathbf{I}_t$ and the two previous frames $\mathbf{I}_{t-1}$ and $\mathbf{I}_{t-2}$ along with the corresponding IMU readings.

(2) Check whether this triplet should be added to the replay buffer using the proposed diversity-based update mechanism.

(3) Sample from the replay buffer and combine the samples with the previously generated data triplet.

(4) Estimate the depth map $\mathbf{D}_{t-1}$ and the camera motions $\mathbf{O}_{t-2 \to t-1}$ and $\mathbf{O}_{t-1 \to t}$.

(5) Compute the loss defined in Eq. (1) and update the network weights via backpropagation.

(6) Repeat steps (4) and (5) for c iterations.

(7) Output $\mathbf{O}_{t-1 \to t}$ as the odometry estimate.

In the following, we provide more details on the proposed replay buffer and the online continual learning strategy of CoVIO. Finally, we propose an asynchronous version of CoVIO that separates the motion estimation from the network update step allowing continuous inference.

3.2.1 Replay Buffer

As outlined in Sec. 1, previous works [10, 30] typically assumed an infinitely sized replay buffer without considering the limited storage capacity on robotic platforms or mobile devices. To address this issue, we use a replay buffer with a fixed maximum size and propose an image diversity-based update mechanism that is comprised of two steps shown in Fig. 3. First, determine whether to add the current online data into the replay buffer and, second, if adding the data results in exceeding the predefined buffer size, select a sample that will be removed from the buffer.

Inspired by the loop closure detection in visual SLAM [12, 30], we interpret the cosine similarity between image feature maps as a distance measure between two frames $\mathbf{I}_1$ and $\mathbf{I}_2$:

$$\text{sim}_{\text{cos}} = \cos\left(\text{feat}(\mathbf{I}_1), \text{feat}(\mathbf{I}_2)\right), \tag{2}$$

where $\text{feat}(\cdot)$ denotes the respective image features. In order to determine whether adding a new sample would increase the diversity of the replay buffer, we compute its cosine similarity with respect to all samples that are already in the buffer and take the maximum value.

$$\text{sim}_{\mathbf{B}}(\mathbf{I}_t) = \max_{\mathbf{I}_i \in \mathbf{B}} \cos\left(\text{feat}(\mathbf{I}_t), \text{feat}(\mathbf{I}_i)\right), \tag{3}$$

where $\mathbf{I}_i \in \mathbf{B}$ refers to the current content of the buffer. If the cosine similarity is below a given threshold, i.e., $\text{sim}_{\mathbf{B}}(\mathbf{I}_t) < \theta_{th}$, the data triplet associated with $\mathbf{I}_t$ is added to the replay buffer. In case this results in a buffer size larger than the allowed size, we have to remove a sample from the buffer. Instead of using random sampling, we remove the sample that yields maximal diversity within the remaining samples. Formally, we remove the following sample:

$$\arg\max_{\mathbf{I}_i \in \mathbf{B}} \sum_{\mathbf{I}_j \in \mathbf{B}} \cos\left(\text{feat}(\mathbf{I}_i), \text{feat}(\mathbf{I}_j)\right) \tag{4}$$

As described in the next section, we do not update the encoder weights of CoVIO. Therefore, to avoid the overhead of a separate network, we use the encoder of the DepthNet to generate image features.
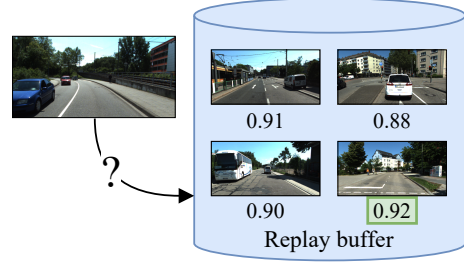
(a) An image is added to the replay buffer if the cosine similarity to the most similar image in the buffer is below a threshold, e.g., $\theta_{th} = 0.95$. Here, the image will be added since $0.92 < \theta_{th}$.
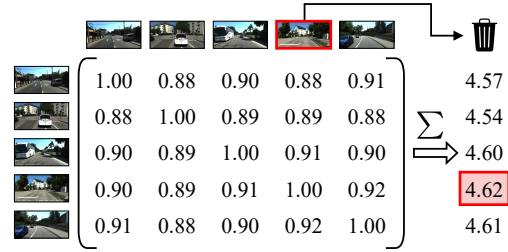
(b) If adding a new image results in exceeding the allowed size of the replay buffer, the image that is the most similar with respect to all other images is removed. The table shows the cosine distance between two frames.

Figure 3. Diversity-based update mechanism of the replay buffer, separated in (a) adding and (b) removing a sample.

3.2.2 Adaptive Visual-Inertial Odometry

After the replay buffer has been updated, we construct a batch $\mathbf{b}_t$ consisting of the data triplet of the current image $\mathbf{I}_t$ and N samples from the replay buffer.

$$\mathbf{b}_t = \{\mathbf{I}_t, \mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_N\} \tag{5}$$

To query the samples from the buffer, we use a uniform probability distribution across all samples and avoid selecting the same sample multiple times if the current size of the buffer is greater than the requested number of samples. To further increase diversity, we augment the replay images in terms of brightness, contrast, saturation, and hue value. Next, the batch $\mathbf{b}_t$, comprising RGB images and velocity measurements, is fed to the DepthNet to estimate a dense depth map of the center images and to the PoseNet to estimate the camera motion with respect to both neighboring frames. Following the same procedure as during pretraining (see Sec. 3.1), we then compute the loss $\mathcal{L}$ defined in Eq. (1) and perform backpropagation to update the network weights. Following McCraith et al. [22], we do not update the weights of the encoders but only of the decoders.
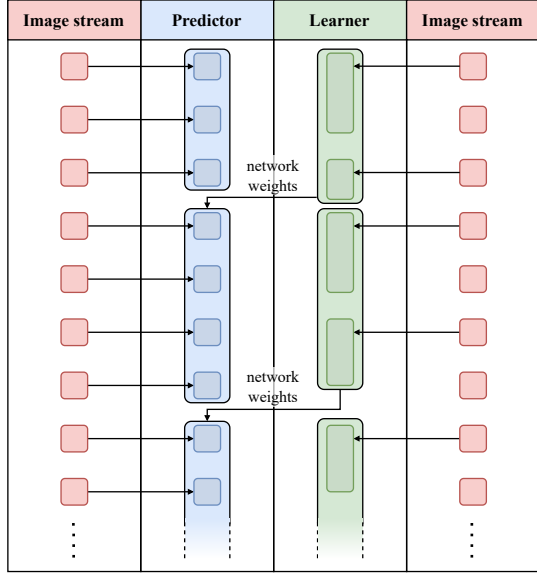
Figure 4. Illustration of the asynchronous variant of CoVIO. While the predictor generates visual odometry estimates in real time, the learner updates the network weights via backpropagation. After a given number of update cycles, the network weights are transferred from the learner to the predictor.

3.2.3 Asynchronous CoVIO

Finally, we propose an asynchronous variant of CoVIO to address true continuous inference on robotic platforms in a real-time capable setting. Since multiple update iterations c can result in a situation, in which the network update takes longer than the frame rate of the input camera stream, we also design a version that decouples the VO estimation from the CL updates. As illustrated in Fig. 4, the *predictor* continuously generates VO estimates for each incoming image. The *learner* contains a copy of the network that is updated using the previously introduced online CL strategy but disregards images if the update step takes longer than the time until the next frame is available. Compared to caching frames, this strategy ensures that always the latest information is used to update the network. Then, after a given number of update cycles, the network weights are transferred from the *learner* to the *predictor*. We include implementations in both ROS and ROS2 in our published code base.

4. Experimental Evaluation

In this section, we present extensive experimental results on the efficiency and efficacy of our proposed CoVIO, compared to previous works. We further conduct multiple ablation studies to demonstrate the effect of newly introduced hyperparameters and to justify certain design choices.

Throughout all experiments, we report the translation error t_{err} (in %) and the rotation error r_{err} (in °/m) as pro-

posed by Geiger *et al*. [5]. These metrics evaluate the error as a function of the trajectory length. To ensure a fair comparison with the base work CL-SLAM [30], we further utilize the set of network weights that is provided by the authors and was pretrained on the Cityscapes Dataset [3]. We also follow CL-SLAM and only consider new frames when the IMU measures a driven distance of at least $0.2\,\mathrm{m}$.

4.1. Datasets

We employ our method on various datasets simulating a diverse set of environments. In particular, we initialize CoVIO with network weights trained on Cityscapes [3] and perform online continual learning on sequences from the KITTI odometry benchmark [5], the Oxford RobotCar Dataset [20], and in-house data.

Cityscapes: The Cityscapes Dataset [3] is a large-scale autonomous driving dataset that contains RGB images and vehicle metadata such as velocity. It was recorded in 50 cities in Germany, France, and Switzerland. In this work, we use network weights pretrained on Cityscapes that are provided by Vödisch *et al*. [30].

KITTI: The KITTI Dataset [5] is a pioneering autonomous driving dataset that was recorded in Karlsruhe, Germany. For continual learning of new domains, we use images and ground truth poses of multiple sequences from the odometry benchmark and combine them with the respective IMU data from the raw dataset.

Oxford RobotCar: The Oxford RobotCar Dataset [20] provides multiple recordings of the same route that were captured across one year. We use the included RGB images and the IMU data. To compute the error metrics, we exploit the separately released RTK ground truth positions [19].

In-House: Finally, we employ CoVIO on an in-house dataset recorded in Freiburg, Germany. Our robotic platform includes forward-facing RGB cameras and an inertial navigation system (INS) that we use to compute the velocity supervision loss.

4.2. Evaluation of Online Continual Learning

In this section, we conduct a series of experiments including both simple online domain adaptation from a source $\mathcal{S}$ to a target domain $\mathcal{T}$ and online continual learning from $\mathcal{S}$ to a sequence of target domains $\{\mathcal{T}_1, \mathcal{T}_2, \dots\}$. Based on the ablation studies in Sec. 4.3, we use a replay buffer size of $|\mathbf{B}| = 100$, an update batch size of $|\mathbf{b}_t| = 3$, and $c = 5$ backpropagation steps allowing a fair comparison with the base work CL-SLAM [30]. We set a similarity threshold of $\theta_{th} = 0.95$ for the diversity-based update scheme of the replay buffer. For the loss weights, we follow CL-SLAM and use $\gamma = 0.001$ and $\lambda = 0.05$. To compare with other methods, we do not use the asynchronous version but the same learning scheme as in CL-SLAM.

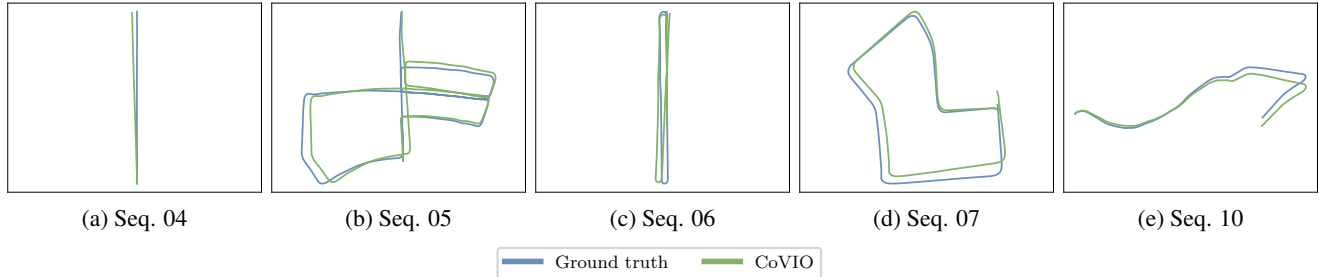| (a) Seq. 04 | (b) Seq. 05 | (c) Seq. 06 | (d) Seq. 07 | (e) Seq. 10 |

— Ground truth   — CoVIO

Figure 5. Online continual learning results on the KITTI odometry benchmark after pretraining on the Cityscapes dataset.

Table 1. Comparison of continual odometry estimation on the KITTI odometry benchmark.

Method	Seq. 04		Seq. 05		Seq. 06		Seq. 07		Seq. 10	
	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}
ORB-SLAM [23]	0.62	0.11	2.51	0.25	7.80	0.35	1.53	0.35	2.96	0.52
Only target	10.72	1.69	34.55	11.88	15.20	5.62	12.77	6.80	55.27	9.50
DeepSLAM [12]	5.22	2.27	_4.04_	1.40	5.99	1.54	4.88	2.14	_10.77_	4.45
Only source	28.94	4.64	46.13	19.20	49.57	20.79	37.75	25.42	30.91	15.28
CL-SLAM [30]	_4.37_	**0.51**	4.30	_1.01_	_2.53_	_0.63_	**2.10**	**0.83**	11.18	_1.74_
CoVIO *(ours)*	**2.11**	_0.53_	**2.88**	**0.94**	**2.13**	**0.47**	_3.19_	_1.26_	**3.71**	**1.55**

Comparison of the translation and rotation errors of our CoVIO with baseline methods evaluated on the KITTI odometry benchmark. "Only target" and DeepSLAM are trained on sequences {00, 01, 02, 08, 09} without further adaptation. "Only source", CL-SLAM, and CoVIO are trained on Cityscapes. Both CL-SLAM and CoVIO perform online adaptation on the respective KITTI sequence. The values of CL-SLAM and "only target" are reported by Vödisch *et al.* [30]. The errors of the paths predicted by ORB-SLAM are based on ground truth scaling and are hence not directly comparable to the other methods. The smallest and second smallest errors across the methods producing metric predictions are shown in **bold** and underlined, respectively.

4.2.1   Cityscapes to KITTI

We use our proposed CoVIO to perform online continual adaptation from Cityscapes to KITTI and compare its performance to other methods shown in Tab. 1. In detail, we compare with the traditional ORB-SLAM [23] as well as the following learning-based methods: "Only target" and DeepSLAM [12] are trained on the KITTI sequences {00, 01, 02, 08, 09} without further adaptation; "only source", CL-SLAM [30], and CoVIO are trained on Cityscapes with online adaptation to KITTI for both CL-SLAM and CoVIO. Generally, the difference between "only source" and "only target" demonstrates the domain gap that online adaptation aims to overcome. Our proposed CoVIO outperforms the base method CL-SLAM on the majority of sequences and also improves performance compared to offline training on the target domain. We visualize the predicted and ground truth odometry in Fig. 5. Note that, unlike CL-SLAM and DeepSLAM, we do not include loop closures in CoVIO.

We further perform online continual learning on all sequences in a sequential manner, *i.e.*, after pretraining on Cityscapes, adapt to sequence 04, then sequence 05, etc., and list the results in Tab. 2. In particular, we compute the translation and rotation errors after each step on all sequences to determine both forward and backward transfer, *i.e.*, the effect on previous and yet unseen future sequences.

Since all the sequences of a dataset could be considered to represent similar domains, *e.g.*, the same camera parameters and comparable environments, we observe a general trend of positive forward transfer. Although the performance on previous sequences cannot be fully retained, CoVIO successfully mitigates catastrophic forgetting compared to the initial performance after pretraining on the source domain.

4.2.2   Cityscapes to In-House Data

Next, we utilize CoVIO to estimate visual odometry on an in-house dataset, after pretraining on Cityscapes. In Fig. 6, we provide a qualitative comparison of CoVIO to CL-SLAM [30] with disabled loop closure detection, no online adaptation, and the measured GNSS position. Since we do not have access to highly accurate RTK readings, we omit computing error metrics for this dataset. However, as demonstrated in Fig. 6, CoVIO is able to maintain accurate odometry tracking for a longer distance than CL-SLAM.

4.2.3   Cityscapes to KITTI and RobotCar

We further investigate the capability of CoVIO to retain knowledge in a multi-target setting. In detail, we perform the same experiment as conducted by CL-SLAM [30]. After initialization on Cityscapes, we sequentially deploy

Table 2. Continual odometry estimation results on the KITTI odometry benchmark.

Sequence	Images	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}
		Cityscapes	$\longrightarrow$	Seq. 04	$\longrightarrow$	Seq. 05	$\longrightarrow$	Seq. 06	$\longrightarrow$	Seq. 07	$\longrightarrow$	Seq. 10	
Seq. 04	269	28.94	4.64	2.11	0.53	7.66	7.05	8.21	1.48	7.88	3.41	9.80	3.82
Seq. 05	2676	46.13	19.20	59.51	16.99	2.85	1.05	8.49	3.77	6.84	3.64	13.73	5.36
Seq. 06	1099	49.57	20.79	65.39	22.33	20.01	10.83	3.08	1.16	7.77	4.25	5.76	1.92
Seq. 07	993	37.75	25.42	67.67	29.85	7.26	4.93	7.13	3.38	6.05	3.53	9.60	5.33
Seq. 10	1127	30.91	15.28	35.37	10.18	11.13	9.48	5.08	2.47	17.53	7.73	2.65	1.15

We continually employ CoVIO on five KITTI sequences after initialization on Cityscapes. The number of images corresponds to the number of update batches of a sequence. The cells highlighted in gray denote the results of the current adaptation step. Along one row, we can measure forward and backward transfer.
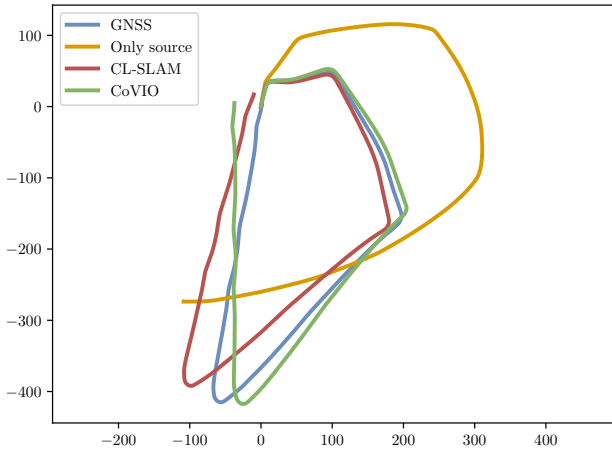

Figure 6. Continual odometry estimation results on in-house data.

CoVIO to KITTI sequence 09, a sequence from RobotCar, KITTI sequence 10, and another sequence from RobotCar. For further details on the RobotCar sequences, we refer the reader to [30]. In Tab. 3, we report the adaptation quality (AQ) and the retention quality (RQ) as introduced by Vödisch *et al.* [30]. Broadly, the AQ score measures the ability of a method to adapt to a previously unseen environment, whereas the RQ measures the ability to retain long-term knowledge when being redeployed to a previously seen domain. Compared to CL-SLAM, CoVIO improves the AQ score and, with a high margin, the RQ with respect to the translation error. Although RQ_{rot} suffers from a small decrease, the absolute rotation errors on the four considered sequences are smaller than those of CL-SLAM, hence smaller differences between with and without an intermediate domain influence the RQ more strongly.

4.3. Ablation Study

In this section, we present the results of various ablation studies substantiating the design choices for the sizes of the update batch b_t and the replay buffer B. We further demonstrate that CoVIO is less sensitive to the number of back-propagation steps per update batch than a previous method.

Table 3. Comparison of adaptation and retention quality.

Previous sequences	Current sequence	CL-SLAM [30] t_{err}	r_{err}	CoVIO *(ours)* t_{err}	r_{err}
c_t	k_9	**2.50**	**0.37**	3.89	1.49
c_t	r_1	28.94	5.63	**6.62**	**2.61**
$c_t \rightarrow r_1$	k_9	**3.24**	**0.54**	4.09	1.18
$c_t \rightarrow k_9$	r_1	30.13	5.87	**11.00**	**3.44**
$\Rightarrow$ AQ$_{trans}$ / AQ$_{rot}$		0.85	0.98	**0.94**	**0.99**
$c_t \rightarrow k_9 \rightarrow r_1$	k_{10}	4.85	1.59	**1.86**	**0.70**
$c_t \rightarrow k_9 \rightarrow r_1 \rightarrow k_{10}$	r_2	20.50	4.77	**5.66**	**3.99**
$c_t \rightarrow k_9$	k_{10}	7.48	1.63	**1.43**	**0.73**
$c_t \rightarrow k_9 \rightarrow r_1$	r_2	16.41	4.58	**7.67**	**3.42**
$\Rightarrow$ RQ$_{trans}$ / RQ$_{rot}$ $\times 10^{-3}$		-7.30	**-0.42**	7.89	-1.53

Adaptation quality (AQ) and retention quality (RQ) [30] with respect to the translation and rotation errors. c_t denotes the Cityscapes training set, k_9 and k_{10} refer to sequences 09 and 10 of the KITTI odometry benchmark, and r_1 and r_2 correspond to sequences [30] from Robot-Car. The values of CL-SLAM are reported by the authors. The best scores in each category are shown in **bold**.

In the following studies, we always report the translation and rotation errors of sequences 04 and 06.

4.3.1 Size of the Update Batch

We first investigate the effect of varying sizes of the update batch, *i.e.*, the number of replay samples. In Tab. 4, we list the errors for batch sizes $b_t = \{1, 2, 3, 4, 5\}$ given an unlimited replay buffer. Note that $b = 1$ implies that experience replay is disabled. Therefore, this strategy corresponds to CL-SLAM [30] for source-to-target domain adaptation. Generally, the translation error can be reduced by using replay data, whereas the rotation error is the smallest when only training with the current sample. As we deem the translation error more important in terms of mapping and localization accuracy, we select $b = 3$.

4.3.2 Size of the Replay Buffer

In the next study, we restrict the size of the replay buffer to address both scalability of the method and the lim-

Table 4. Ablation study on the size of the update batch.

Batch size	Seq. 04		Seq. 06	
	t_{err}	r_{err}	t_{err}	r_{err}
1	3.56	**0.15**	2.30	**0.18**
2	2.97	0.59	**1.81**	<u>0.50</u>
3	**2.79**	<u>0.54</u>	<u>1.98</u>	0.59
4	<u>2.89</u>	0.73	1.99	0.54
5	<u>2.89</u>	0.63	2.46	0.70

In this study, we use a replay buffer of infinite size. Batch sizes greater than 1 imply using replay data in addition to the online image, *i.e.*, the first row corresponds to the strategy of CL-SLAM. The smallest and second smallest errors are shown in **bold** and <u>underlined</u>, respectively.

Table 5. Ablation study on the size of replay buffer.

Buffer size	Diversity update	Seq. 04		Seq. 06	
		t_{err}	r_{err}	t_{err}	r_{err}
∞		2.79	0.54	1.98	0.59
100		2.62	0.52	**1.75**	<u>0.48</u>
100	✓	**2.11**	0.53	2.13	**0.47**
50		2.64	0.42	2.72	0.91
50	✓	**2.11**	0.53	2.13	**0.47**
25		2.51	<u>0.40</u>	2.42	0.77
25	✓	**2.11**	0.53	2.20	0.50
10		2.82	**0.33**	<u>2.08</u>	0.64
10	✓	<u>2.12</u>	0.56	2.21	0.59

In this study, we use a batch size of 3. The effectively used buffer size of sequence 04 is the same for 25, 50, and 100. Similarly, in sequence 06 the same number of samples is added when buffer sizes of 50 and 100 are available. The first row corresponds to the strategy of CL-SLAM. The smallest and second smallest errors are shown in **bold** and <u>underlined</u>, respectively.

ited storage capacity on mobile devices and robotic platforms. In detail, we report the errors for buffer sizes $|\mathbf{B}| = \{10, 25, 50, 100, \infty\}$ in Tab. 5. For all the buffers of limited size, we both enable and disable our proposed diversity-based updating mechanism. In the latter case, we sample an element from the replay buffer assuming a uniform distribution and replace this with the current online data. Interestingly, the positive effect of enforcing a high diversity is more pronounced for sequence 04, where CoVIO generally yields smaller translation errors with fewer samples in the buffer. It should further be noted that due to the length of the sequences, the diversity-based buffer contains the same samples for $|\mathbf{B}_{\text{Seq. 04}}| = \{25, 50, 100\}$ and $|\mathbf{B}_{\text{Seq. 06}}| = \{50, 100\}$. For CoVIO, we select $|\mathbf{B}| = 100$ to account for the increased storage requirements in a multi-target setting.

4.3.3 Number of Update Cycles

Lastly, we report the sensitivity of CoVIO with respect to the number of backpropagation steps c for a single up-

Table 6. Ablation study on the number of update cycles.

Update cycles	Seq. 04		Seq. 06	
	t_{err}	r_{err}	t_{err}	r_{err}
1	3.13	1.65	4.07	1.14
2	2.80	1.06	2.62	0.64
3	2.49	0.77	2.61	0.63
4	2.29	0.61	2.31	0.52
5	<u>2.11</u>	<u>0.53</u>	**2.13**	**0.47**
6	**1.98**	**0.50**	<u>2.15</u>	<u>0.49</u>

For a fair comparison, we use the same number of update cycles $c = 5$ as CL-SLAM [30]. The smallest and second smallest errors are shown in **bold** and <u>underlined</u>, respectively.

date batch. As shown in Tab. 6, more steps decrease both the errors confirming the results of CL-SLAM [30]. However, in contrast to the performance reported for CL-SLAM, CoVIO is noticeably less sensitive and already yields relatively small errors for $c = 1$. We conclude that this is caused by using experience replay also for the online learner. To enable direct comparison with CL-SLAM, we use the same value $c = 5$.

5. Conclusion

In this paper, we presented CoVIO for online continual learning of visual-inertial odometry. CoVIO exploits an unsupervised training scheme during inference time and thus seamlessly adapts to new domains. In particular, we explicitly address the shortcomings of previous works by designing a lightweight network architecture and by proposing a novel fixed-size replay buffer that maximizes image diversity. Using experience replay, CoVIO successfully mitigates catastrophic forgetting while achieving efficient online domain adaptation. We further provide an asynchronous version of CoVIO separating the core motion estimation from the network update step, hence allowing true continuous inference in real time. In extensive evaluations, we demonstrate that CoVIO outperforms competitive baselines. We also make the code and models publicly available to facilitate future research. Future work will focus on extending this work to a multi-task setup for robotic vision systems.

References

[1] 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. 1

[2] 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. 1, 2

[3] 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. 5

[4] Changhong Fu, Adrian Carrio, and Pascual Campoy. Efficient visual odometry and mapping for unmanned aerial vehicle using arm-based stereo vision pre-processing system. In *International Conference on Unmanned Aircraft Systems*, pages 957–962, 2015. 1

[5] 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. 5

[6] 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. 1, 2, 3

[7] Vitor Guizilini, Rareș Ambruș, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3D packing for self-supervised monocular depth estimation. In *IEEE/CVF Conference Computer Vision and Pattern Recognition*, 2020. 3

[8] 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. 3

[9] Pan Ji, Qingan Yan, Yuxin Ma, and Yi Xu. Georefine: Self-supervised online depth refinement for accurate dense mapping. In *European Conference on Computer Vision*, pages 360–377, 2022. 2

[10] 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*, 2021. 2, 4

[11] 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. 2

[12] Ruihao Li, Sen Wang, and Dongbing Gu. DeepSLAM: A robust monocular SLAM system with unsupervised deep learning. *IEEE Transactions on Industrial Electronics*, 68(4):3577–3587, 2021. 4, 6

[13] 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*, 2020. 1, 2

[14] Shunkai Li, Xin Wu, Yingdian Cao, and Hongbin Zha. Generalizing to the open world: Deep visual odometry with online adaptation. In *IEEE/CVF Conference Computer Vision and Pattern Recognition*, pages 13179–13188, 2021. 2

[15] Shing Yan Loo, Moein Shakeri, Sai Hong Tang, Syamsiah Mashohor, and Hong Zhang. Online mutual adaptation of deep depth prediction and visual slam. *arXiv preprint arXiv:2111.04096*, 2021. 2

[16] David Lopez-Paz and Marc' Aurelio Ranzato. Gradient episodic memory for continual learning. In *Conference on Neural Information Processing Systems*, volume 30, 2017. 2

[17] Mayank Lunayach, James Smith, and Zsolt Kira. Lifelong wandering: A realistic few-shot online continual learning setting. *arXiv preprint arXiv:2206.07932*, 2022. 2

[18] Hongcheng Luo, Yang Gao, Yuhao Wu, Chunyuan Liao, Xin Yang, and Kwang-Ting Cheng. Real-time dense monocular SLAM with online adapted depth prediction network. *IEEE Transactions on Multimedia*, 21(2):470–483, 2019. 1, 2

[19] Will Maddern, Geoffrey Pascoe, Matthew Gadd, Dan Barnes, Brian Yeomans, and Paul Newman. Real-time kinematic ground truth for the oxford robotcar dataset. *arXiv preprint arXiv:2002.10152*, 2020. 5

[20] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000km: The Oxford RobotCar dataset. *International Journal of Robotics Research*, 36(1):3–15, 2017. 5

[21] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3):169–186, 2007. 1

[22] Robert McCraith, Lukas Neumann, Andrew Zisserman, and Andrea Vedaldi. Monocular depth estimation with self-supervised instance adaptation. *arXiv preprint arXiv:2004.05821*, 2020. 4

[23] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 6

[24] Thomas Schöps, Jakob Engel, and Daniel Cremers. Semi-dense visual odometry for ar or on a smartphone. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 145–150, 2014. 1

[25] Abu Md Niamul Taufique, Chowdhury Sadman Jahan, and Andreas Savakis. Unsupervised continual learning for gradually varying domains. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 3739–3749, 2022. 2

[26] Sebastian Thrun. Is learning the n-th thing any easier than learning the first? In *Conference on Neural Information Processing Systems*, volume 8, 1995. 2

[27] Abhinav Valada, Noha Radwan, and Wolfram Burgard. Deep auxiliary learning for visual localization and odometry. In *IEEE International Conference on Robotics and Automation*, pages 6939–6946, 2018. 1

[28] Abhinav Valada, Noha Radwan, and Wolfram Burgard. Incorporating semantic and geometric priors in deep pose regression. *Workshop on learning and inference in robotics: Integrating structure, priors and models at robotics: Science and systems (RSS)*, 1(3), 2018. 1

[29] Francisco Rivera Valverde, Juana Valeria Hurtado, and Abhinav Valada. There is more than meets the eye: Self-supervised multi-object detection and tracking with sound by distilling multimodal knowledge. In *IEEE/CVF Conference Computer Vision and Pattern Recognition*, pages 11607–11616, 2021. 2

[30] 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. 1, 2, 3, 4, 5, 6, 7, 8

[31] Niclas Vödisch, Kürsat Petek, Wolfram Burgard, and Abhinav Valada. CoDEPS: Online continual learning for depth estimation and panoptic segmentation. *arXiv preprint arXiv:2303.10147*, 2023. 2

[32] Jianren Wang, Xin Wang, Yue Shang-Guan, and Abhinav Gupta. Wanderlust: Online continual object detection in the real world. In *International Conference on Computer Vision*, pages 10809–10818, 2021. 2